

UNIVERSIDAD POLITÉCNICA DE VALENCIA



DEPARTAMENTO DE INFORMÁTICA DE  
SISTEMAS Y COMPUTADORES

**Reserva Eficiente de Recursos en Redes para  
Transmisión en Tiempo Real**

TESIS DOCTORAL PRESENTADA POR:

*Enrique Hernández Orallo*

DIRIGIDA POR:

*Joan Vila i Carbó*

Valencia, enero de 2001



A mi mujer Elvira,  
a mi hijo David y al que viene,  
por su apoyo y cariño.



# Resumen

*La demanda de calidad de servicio y transmisión en tiempo real se ha incrementado debido al desarrollo de nuevas aplicaciones multimedia. Esta transmisión de información multimedia implica un gran consumo de recursos en las redes de comunicación. Esto es debido fundamentalmente a las características de este tipo de tráfico y a la calidad de servicio requerida por las aplicaciones que lo necesitan.*

*Normalmente, para la realización de esta transmisión, se establece un canal. El establecimiento de este canal implica la reserva de unos recursos de red (normalmente ancho de banda y memoria) que dependen principalmente de las características del tráfico pero también de los parámetros de la red.*

*El tráfico multimedia suele tener alta variabilidad en la tasa de transmisión, lo que implica que para garantizar la calidad de servicio se hagan unas reservas elevadas de recursos. Pero debido a la alta variabilidad de este tráfico, en muchos momentos estos recursos no son utilizados. La disminución de la reserva puede implicar un incumplimiento de la calidad de servicio especificado.*

*Por lo tanto, un aspecto clave en estas redes es la optimización de la reserva de recursos. Se requieren por tanto, métodos con los que poder estimar esta reserva a partir de las características del tráfico y la red. Estos métodos tienen que ser eficientes y rápidos para que se puedan ejecutar en tiempo de transmisión.*

*En esta tesis se introducen varios esquemas para el ahorro de recursos en la red. El primero de ellos está basado en condensar la información de un tráfico de tasa variable en una serie de puntos. Estos puntos son la base de un nuevo método para obtener la reserva óptima de ancho de banda en los enlaces así como de memoria en los nodos que por su rapidez puede ser usado en el momento del establecimiento del canal. Para ambos casos se utiliza una especificación de tráfico leaky o token bucket. Como punto final se propone un nuevo esquema de activación de canales de reserva con el objetivo de minimizar la reserva conjunta de recursos del canal primario y secundario. Todos estos esquemas han sido probados usando tráfico MPEG.*



# Abstract

*The demand of Quality of Service (QoS) and real-time channels on computer networks has recently increased due to the development of new multimedia applications.*

*This multimedia transmission requires a very demanding reservation of network resources. This is due mainly to the traffic characteristics and the quality of service the applications require.*

*This transmission is usually done using a channel. Channel establishment requires reserving some network resources (bandwidth and buffers) in order to implement the real-time channel. This reservation mainly depends on traffic characteristics, but also on network parameters.*

*Multimedia traffic usually has a very variable transmission rate (VBR). That implies that in order to guarantee the quality of service, the reservation of network resources is very high. However, these resources are usually misused when transmission rate is low.*

*Therefore, the optimisation of network resources is a key issue. There is a need for methods that obtain the optimal reservation of network resources which guarantee quality of service. These methods must be efficient enough in order to be used on-line*

*In this dissertation, some new schemes to save network resources are introduced. The first method is based on generating a set of points from the stored video with an off-line analysis of its empirical envelope, and then using these points to efficiently calculate the optimal bandwidth reservation for a given channel, at channel establishment time. These points are used to optimise the buffer node. In both cases, a leaky or token bucket flow specification is used. Finally, a new scheme for fault detection for highly available real-time channels is developed. This new scheme provides an efficient resource reservation and it is based on “suspecting” a failure when the packet delay through the primary channel is close to the maximum guaranteed delay. These schemes have been proved using MPEG traffic traces.*



# Índice General

<b>Resumen .....</b>	<b>v</b>
<b>Abstract .....</b>	<b>vii</b>
<b>Índice General.....</b>	<b>ix</b>
<b>Objetivos y contribuciones.....</b>	<b>xv</b>
Objetivos.....	xv
Estado del arte .....	xv
Contribuciones.....	xvi
<b>Organización de la tesis.....</b>	<b>xix</b>
<b>1. Introducción.....</b>	<b>1</b>
1.1.- Introducción a la transmisión en tiempo real. ....	1
1.2.- Tráfico multimedia.....	3
1.2.1.- Ancho de banda.....	3
1.2.2.- Retraso de transmisión.....	3
1.2.3.- Fiabilidad.....	4
1.2.4.- Sincronización de canales y multidifusión.....	5
1.3.- Calidad de servicio .....	5
1.3.1.- Aplicaciones de tiempo real.....	6
1.3.2.- Aplicaciones elásticas .....	7
1.3.3.- Ancho de banda y retraso .....	7
1.3.4.- Nivel de calidad de servicio .....	7
1.4.- Modelos de servicios.....	8
1.4.1.- Modelo de servicio integrados .....	8
1.4.2.- Modelo de servicios diferenciados.....	9
1.4.3.- Tráfico agregado y por conversación.....	9

1.4.4.- <i>Requerimientos para compartir recursos</i> .....	10
<b>2. Transmisión en tiempo real</b> .....	<b>11</b>
2.1.- <i>Introducción</i> .....	11
2.2.- <i>Componentes comunes para la transmisión en tiempo real</i> .....	12
2.3.- <i>Modelo conceptual de una red en tiempo real</i> .....	14
2.4.- <i>Especificación y conformación del tráfico</i> .....	15
2.4.1.- <i>Algoritmo leaky bucket</i> .....	16
2.4.2.- <i>Algoritmo token bucket</i> .....	16
2.4.3.- <i>Modelo Tenet</i> .....	17
2.5.- <i>Planificadores. Disciplinas de servicio</i> .....	18
2.5.1.- <i>Introducción</i> .....	18
2.5.2.- <i>Disciplinas de servicio</i> .....	19
2.5.3.- <i>Servicio RCSP</i> .....	20
2.5.4.- <i>Planificador WFQ</i> .....	21
2.6.- <i>Gestión del tráfico: control de admisión</i> .....	23
2.7.- <i>Protocolos de comunicación con calidad de servicio</i> .....	24
2.8.- <i>Calidad de servicio en ATM</i> .....	25
2.8.1.- <i>Introducción</i> .....	25
2.8.2.- <i>Componentes de la gestión del tráfico</i> .....	26
2.8.3.- <i>El contrato del tráfico</i> .....	27
2.8.4.- <i>Categorías de servicio</i> .....	27
2.8.5.- <i>Parámetros de calidad de servicio</i> .....	28
2.8.6.- <i>Modelo del tráfico</i> .....	29
2.8.7.- <i>Ajuste del tráfico</i> .....	30
2.8.8.- <i>Control de admisión de conexiones</i> .....	31
2.9.- <i>Tenet suite</i> .....	33
2.9.1.- <i>Introducción</i> .....	33
2.9.2.- <i>El diseño de los protocolos Tenet</i> .....	33
2.9.3.- <i>Especificación del tráfico</i> .....	34
2.9.4.- <i>Disciplina de servicio</i> .....	35
2.9.5.- <i>Control de admisión</i> .....	36
2.10.- <i>RSVP (Resource ReSerVation Protocol)</i> .....	39
2.10.1.- <i>Introducción</i> .....	39

2.10.2.- <i>Objetivos de diseño</i> .....	40
2.10.3.- <i>Principios de diseño</i> .....	40
2.10.4.- <i>Clases de calidad de servicio</i> .....	42
2.10.5.- <i>Funcionamiento de RSVP</i> .....	44
2.10.6.- <i>Control de admisión</i> .....	47
2.10.7.- <i>Modelos de reserva de recursos</i> .....	48
2.10.8.- <i>Tipos de encaminamiento para RSVP</i> .....	49
2.11.- <i>Arquitectura de servicios diferenciados</i> .....	50
2.12.- <i>Conclusiones</i> .....	53
<b>3. Descripción del tráfico</b> .....	<b>55</b>
3.1.- <i>Características del tráfico</i> .....	55
3.1.1.- <i>Introducción</i> .....	55
3.1.2.- <i>Codificación y compresión</i> .....	56
3.1.3.- <i>Tráfico en ráfagas y dependencias a largo plazo</i> .....	57
3.2.- <i>Modelado del tráfico</i> .....	57
3.2.1.- <i>Concepto de modelo de tráfico</i> .....	57
3.2.2.- <i>Modelos de tráfico</i> .....	58
3.3.- <i>Suavizado</i> .....	59
3.3.1.- <i>Técnicas de suavizado</i> .....	59
3.3.2.- <i>Suavizado en la red</i> .....	60
3.4.- <i>Envolvente empírica</i> .....	62
3.5.- <i>Comparación de tráfico</i> .....	64
3.5.1.- <i>Introducción</i> .....	64
3.5.2.- <i>Funciones de tráfico limitado</i> .....	65
3.5.3.- <i>Obtención de los parámetros</i> .....	68
3.6.- <i>Conclusiones</i> .....	71
<b>4. Reserva óptima de ancho de banda</b> .....	<b>73</b>
4.1.- <i>Introducción</i> .....	73
4.1.1.- <i>Antecedentes y resultados previos</i> .....	75
4.1.2.- <i>Objetivo de la optimización</i> .....	76
4.2.- <i>Cálculo del retraso emisor - receptor</i> .....	76
4.3.- <i>Optimización de la reserva de ancho de banda</i> .....	78
4.4.- <i>Descripción de un método para la reserva óptima en-línea</i> .....	84

4.4.1.- <i>Introducción</i> .....	84
4.4.2.- <i>Cálculo de los puntos envolventes</i> .....	85
4.4.3.- <i>Obtención de la reserva óptima</i> .....	88
4.5.- Algunos resultados sobre tráficos MPEG .....	92
4.6.- Conclusiones .....	96
<b>5. Optimización de la memoria en los nodos.....</b>	<b>97</b>
5.1.- <i>Introducción</i> .....	97
5.2.- <i>Cálculo del uso de memoria en los nodos</i> .....	98
5.3.- <i>Optimización de la reserva</i> .....	100
5.4.- <i>Método para la optimización de la memoria en un nodo</i> .....	102
5.5.- <i>Implementación</i> .....	105
5.6.- <i>Resultados comparativos y pruebas</i> .....	108
5.6.1.- <i>Resultados con dos clases</i> .....	110
5.6.2.- <i>Resultados con cinco cargas</i> .....	112
5.6.3.- <i>Conclusiones</i> .....	115
5.7.- <i>Aplicación a varios nodos</i> .....	115
5.7.1.- <i>Conformación del tráfico en la red</i> .....	116
5.7.2.- <i>Caracterización de la distorsión en la red</i> .....	119
5.8.- <i>Balanceo en la optimización</i> .....	122
5.9.- <i>Conclusiones</i> .....	124
<b>6. Reserva eficiente de recursos para canales de reserva .....</b>	<b>125</b>
6.1.- <i>Introducción</i> .....	125
6.2.- <i>Esquema de detección de fallo</i> .....	126
6.3.- <i>Redes de prueba y cálculo de la reserva</i> .....	129
6.4.- <i>Evaluación analítica del ahorro de recursos</i> .....	131
6.4.1.- <i>Introducción</i> .....	131
6.4.2.- <i>Evaluación para la red 1</i> .....	134
6.4.3.- <i>Evaluación para la red 2</i> .....	137
6.4.4.- <i>Evaluación para la red 3</i> .....	137
6.4.5.- <i>Conclusiones</i> .....	139
6.5.- <i>Resultados de Simulaciones</i> .....	139
6.5.1.- <i>Introducción</i> .....	139
6.5.2.- <i>Resultados para la red 1</i> .....	144

6.5.3.- Resultados para la red 2 .....	144
6.5.4.- Resultados para la red 3 .....	146
6.5.5.- Conclusiones .....	147
6.5.6.- Resultados para leaky bucket.....	147
6.6.- Implementación.....	149
6.7.- Conclusiones .....	153
<b>7. Conclusiones.....</b>	<b>155</b>
7.1.- Conclusiones .....	155
7.1.1.- Estado de las soluciones propuestas.....	155
7.1.2.- Optimización de la reserva .....	156
7.1.3.- Optimización con canales de reserva .....	157
7.1.4.- Simulación y carga.....	157
7.2.- Líneas de trabajo futuras .....	158
<b>Apéndice 1: Trazas MPEG usadas .....</b>	<b>161</b>
A1.1.- Introducción.....	161
A1.2.- Codificación de vídeo MPEG.....	161
A1.3.- Descripción de los tráficoes .....	163
A1.4.- Conclusiones .....	172
<b>Apéndice 2: Descripción de RTNetSim .....</b>	<b>173</b>
A2.1.- Introducción.....	173
A2.2.- Simulador RTNetSim .....	174
A2.3.- Estructura global de RTNetSim .....	175
A2.4.- Especificación de la red y carga.....	176
A2.5.- Estructura de la red.....	181
A2.6.- Resultados y estadísticas .....	181
A2.7.- Conclusiones .....	183
<b>Apéndice 3: Modelo de red ORNet .....</b>	<b>185</b>
A3.1.- Introducción.....	185
A3.2.- Arquitectura de la red. ....	186
A3.3.- Interfaz de programación.....	188
A3.4.- Comportamiento dinámico .....	189
A3.5.- Conclusiones .....	192

**Apéndice 4: Acrónimos..... 193**

**Bibliografía ..... 195**

# Objetivos y contribuciones

## *Objetivos*

El objetivo principal de esta tesis, como refleja el título, es la reserva de recursos de forma eficiente en redes en tiempo real. Como se ha comentado en el resumen, la base para poder garantizar una calidad de servicio en redes en tiempo real es la reserva de sus recursos. Normalmente, esta reserva es muy elevada lo cual limita la capacidad de las redes para admitir muchos canales sin una merma en la calidad del servicio. Cualquier reducción en esta reserva, garantizando siempre la calidad de servicio deseada, implicará que las redes se utilicen mejor y permitan admitir más canales.

Se está hablando por tanto, de un problema de eficiencia en el uso de unos recursos compartidos y de alto coste como suelen ser los de las redes de comunicaciones.

## *Estado del arte*

El concepto de transmisión en tiempo real fue establecido por Doménico Ferrari [Ferrari89] a finales de la década de los 80. El grupo de investigación sobre transmisión en tiempo real *Tenet* de la *University of California at Berkeley* creado por Ferrari fue muy activo en los primeros años de la década de los 90. Los trabajos creados por este grupo están basados en aplicar los conceptos tradicionales de tiempo real (como planificadores con prioridades, tareas planificables, etc.) a redes en tiempo real. Todas estas ideas fueron plasmadas en el diseño e implementación de una red en tiempo real denominada *Tenet suite* [Banerjea96]. Este grupo estableció conceptos tan importantes como la función de tráfico limitado, envolvente empírica y la caracterización del tráfico [Knightly94].

A principios de los 90 se empezó a aplicar a las redes en tiempo real la teoría de flujo. En este sentido, dos trabajos fueron básicos para aplicar este modelo y poder obtener retrasos acotados: el modelo matemático para el cálculo de retraso en redes presentado por R. Cruz en 1991 [Cruz91a][Cruz91b] y la obtención de una ecuación del retraso con redes con planificadores WFQ (*Weighted Fair Queuing*) presentado por Parekh y Gallager en

1992 [Parekh92] [Parekh93]. A partir de estos trabajos se han planteado muchas variantes sobre los planificadores y redes de transmisión.

Como consecuencia de todos estos desarrollos, la IETF (*Internet Engineering Task Force*) creó alrededor de 1993 varios grupos de trabajo con el objetivo de proporcionar calidad de servicio (QoS: *Quality of Service*) a redes IP, entre los que destacan el *Integrated Services Group (IntServ)* y el *Resource Reservation Setup Protocol Group (RSVP)*. El resultado de estos grupos es la especificación de un protocolo de reserva de recursos [Braden97] y varias especificaciones de calidad de servicio [Schenker96][Wroclawski96].

El modelo de servicios diferenciados ha sido utilizado durante algunos años, aunque se ha hecho poco esfuerzo para su estandarización hasta hace poco con la creación de un grupo de trabajo de la IETF denominado *Differentiated Services Group (DiffServ)*. El resultado de este grupo es la creación de una arquitectura para la gestión de servicios diferenciados que es fácilmente integrable en redes IP [Blake98].

Los últimos avances en la transmisión en tiempo real se han centrado principalmente en mejorar distintos aspectos de estas redes. Se han introducido nuevos planificadores (un buen resumen en [Zhang98]), algunos para facilitar su implementación hardware para redes ATM [Varma97][Kweon98] y nuevas especificaciones de tráfico para permitir variaciones durante la transmisión. También se ha trabajado en mecanismos de compresión y codificación de vídeo para la reducción de la reserva (una buena descripción en [Lakshman97]). También se han implementado distintos esquemas para tolerancia a fallos [Banerjee93] [Han97] [Han99].

## Contribuciones

Las principales contribuciones que aporta esta tesis son las siguientes:

- Usando la envolvente empírica se establece un mecanismo para poder comparar distintas especificaciones de tráfico lo cual permite comprobar su eficiencia (capítulo 3).
- Se introduce un nuevo esquema eficiente para la reserva óptima de ancho de banda. Para ello se introduce el concepto de los puntos envolventes generados a partir de envolvente empírica. Estos puntos contienen información condensada que permiten obtener el valor óptimo de la reserva de ancho de banda a realizar entre emisor y receptor. Este nuevo método es muy eficiente (en cinco o seis

iteraciones se encuentra la solución) y por tanto, permite su aplicación para el vídeo bajo demanda (capítulo 4).

- Dado que el número de puntos envolventes es bajo para los tráficos estudiados esta información se puede transmitir a los nodos. Con estos puntos en los nodos se desarrolla un método para optimizar la reserva de memoria que mejora ampliamente el establecido (capítulo 5).
- Siguiendo este esquema se pueden agrupar varios canales en un macrocanal para optimizar la reserva en una subred (capítulo 5).
- Se desarrolla un nuevo esquema de detección de fallos que optimiza la reserva de recursos en los canales de reserva proporcionando garantías sobre el retraso (capítulo 6).
- Para evaluar este esquema se ha desarrollado un simulador de redes en tiempo real: RTNetSim (apéndice 2).

Es importante recalcar que todas estas optimizaciones son fácilmente aplicables a redes en tiempo real que proporcionen modelos de flujo *leaky* o *token bucket* con planificadores del estilo WFQ. En el apéndice 3 se introduce un modelo de red en tiempo real que pueda servir de base para su futura implementación. En él se puede comprobar que no se requiere la modificación de las disciplinas de servicio en los nodos ni el modelo del tráfico. Además, permite su fácil integración en redes complejas con distintas subredes como ATM.

Estas contribuciones se han presentado en varios congresos: el nuevo esquema de detección de fallos para canales de reserva fue presentado en septiembre de 1999 en Madrid (JITEL'99) [Hernández99a] y en noviembre de 1999 en el MIT, Massachusetts, USA (PDCS'99) [Hernández99b]. La evaluación de este nuevo esquema con las pruebas realizadas con el simulador fue presentada en octubre de 2000 en Bergen, Noruega (Interworking'00) [Hernández00b]. El material presentado comprende el capítulo 6 de la tesis.

El nuevo esquema de reserva de recursos óptimo utilizando los puntos envolventes fue presentado en septiembre de 2000 en Maastricht, Holanda (Euromicro 2000) [Hernández00a]. Este material compone íntegramente el cuarto capítulo de la tesis y parte del quinto.



# Organización de la tesis

La tesis está dividida en siete capítulos más tres apéndices. En la primera parte de esta tesis, que comprende los tres primeros capítulos, se centra el problema objeto de esta tesis y se detallan las soluciones aportadas. Esto servirá de base para el resto de los capítulos donde se describen las aportaciones realizadas.

En el primer capítulo se realiza una breve introducción de la transmisión en tiempo real y las características del tráfico multimedia, comentando la problemática asociada. El segundo capítulo describe con más profundidad cuáles son los componentes necesarios para la transmisión en tiempo real y se realiza una breve descripción de algunos protocolos en tiempo real. En el tercer capítulo se describen los tipos de tráfico multimedia y se introduce el concepto de envolvente empírica, que es necesario para el resto de la tesis. A partir de este punto se detallan las contribuciones de la tesis. Por medio de la envolvente empírica se permite comparar distintas especificaciones del tráfico.

En el cuarto capítulo se introduce un nuevo esquema para el cálculo de la reserva óptima basado en el concepto de los puntos envolventes. Se indica su aplicación para el vídeo baja demanda y se realizan estudios sobre tráficos MPEG. En el capítulo quinto se describe un nuevo método para la optimización del *buffer* requerido en un nodo, teniendo en cuenta varios canales y utilizando los puntos envolventes.

En el sexto capítulo se introduce un nuevo esquema para la reserva eficiente de recursos usando canales de reserva, basándose en los resultados de los capítulos anteriores. Finalmente, en el último capítulo se enumeran las conclusiones y se introducen nuevas líneas de trabajo.

En el primer apéndice se muestran los tráficos MPEG usados a lo largo de la tesis. En el segundo se describe el simulador utilizado para probar los distintos esquemas en la tesis. El tercer apéndice detalla un modelo de red en tiempo real: ORNet, que implementa los esquemas de optimización introducidos en la tesis.



# 1.

## Introducción

### *1.1.- Introducción a la transmisión en tiempo real.*

El concepto multimedia, es decir, la integración de los medios, era hasta hace poco una idea remota que se ha convertido con el paso de los años en una realidad. Hasta ahora, es común que toda la información (imágenes, texto, voz, etc.) llegue a nosotros por canales distintos: fotografía, radio, prensa, televisión, cine, etc. Actualmente, es posible que toda esta información multimedia pueda ser accesible a través de las redes de comunicaciones. Esto implica un cambio en la forma de distribución de esta información: un usuario se puede conectar a un servidor para ver la película deseada, acceder a conferencias en tiempo real, operaciones quirúrgicas a distancia, etc.

En otro aspecto, la era de los medios unidireccionales, es decir sólo receptivos, ha dado paso a la era de los medios interactivos: CD-ROM, DVD y Web-TV. Además, gracias al crecimiento de las redes de comunicación, Internet en particular, ya es posible formar parte de un mundo enteramente interactivo y multimedia.

Este incremento de la velocidad y capacidad de los distintos componentes de las redes de comunicaciones así como de los equipos de acceso (ordenadores personales, decodificadores de TV, etc) ha permitido empezar a utilizar nuevos tipos de aplicaciones como la videoconferencia, visualización científica y médica, telecontrol, vídeo y audio bajo demanda, etc., pero no siempre con los resultados deseados. Estas aplicaciones tienen unos requerimientos elevados en términos de ancho de banda, retraso de transmisión permitido y tasa de errores.

La transmisión multimedia, aunque utilizada en Internet con resultados bastantes decepcionantes, va a ser utilizada ampliamente en otros tipos de servicios. Su uso se está extendiendo a la televisión digital por cable, tercera generación de móviles UMTS, radio

digital, videoconferencia, etc. También se prevén nuevas aplicaciones avanzadas como acceso en tiempo real a instrumentación remota, tele-inmersión, realidad virtual, etc.

El esquema de servicio ofrecido por las redes actuales es totalmente inadecuado, por lo que se requieren nuevos servicios que permitan una transmisión en tiempo real. Esto implica nuevos requisitos a la redes transmisión, que se concretan en la especificación de una calidad de servicio (*QoS: Quality of Service*).

En esta tesis se va a utilizar la siguiente definición de transmisión en tiempo real introducida por Ferrari [*Ferrari90b*] <sup>1</sup>:

*“Se denomina servicio de transmisión en tiempo real aquel cuyos clientes pueden especificar los requerimientos sobre prestaciones y obtener garantía del cumplimiento de estos requerimientos.”*

Por tanto, se puede hablar de transmisión de información en tiempo real cuando se puede asegurar que la información llegue a su destino con unos parámetros determinados (retraso, rendimiento, fiabilidad, etc.). En este sentido se puede asumir que la transmisión multimedia tiene unos requerimientos temporales que necesitan del uso de esta transmisión en tiempo real.

Para la transmisión de información en tiempo real se requiere un sistema de comunicaciones que proporcione a las aplicaciones los servicios y el control necesarios para la gestión de la calidad del servicio (*QoS*). En particular, la comunicación de audio o vídeo requiere la provisión de cierto nivel de calidad, garantizándolo durante el tiempo de la transmisión. Este nivel de calidad lo tienen que establecer las aplicaciones a la hora de comunicarse. En general el medio (físico o virtual) por el que se realizan estas comunicaciones se suele denominar canal. Estos canales son definidos con unos parámetros típicos como pueden ser el ancho de banda, retraso, variación del retraso (*delay jitter*) y fiabilidad.

Las redes basadas en paquetes como IP tienen como objetivo maximizar la utilización de la red por medio de la multiplexación de los canales. Además, pueden proporcionar comunicación multipunto y fiabilidad adaptándose a la dinámica de la red. Sin embargo, este funcionamiento hace que el comportamiento sea difícilmente predecible. En cambio las redes basadas en conexión proporcionan un servicio garantizado, pero sin

---

<sup>1</sup> Se ha extendido el uso del término de transmisión en tiempo real como transmisión multimedia, aunque realmente la transmisión multimedia requiere otros parámetros aparte de los meramente temporales. En general, el uso de este término se ha extendido a parámetros como rendimiento o fiabilidad.

embargo, hacen un uso no eficiente de los recursos de la red, no se adaptan a los fallos de la red y no soportan comunicaciones multipunto.

Se puede hablar de una Red de Paquetes de Servicios Integrados (RPSI) en la que se mezclan estos dos paradigmas de comunicación: combinando la comunicación multipunto y multiplexada, la robustez de la red de paquetes conmutados y la garantía de servicio del modelo de conexión.

## ***1.2.- Tráfico multimedia***

Se suele denominar tráfico multimedia aquel compuesto por vídeo y audio, aunque formalmente se refiere a cualquier medio que se puede transmitir. En general, los requerimientos para la transmisión de audio y vídeo han sido extensamente tratados [Hehmann90] y se describen en los siguientes puntos.

### ***1.2.1.- Ancho de banda***

El tráfico multimedia maneja grandes volúmenes de información. Por ejemplo, un CD-ROM, que son 72 minutos de sonido estéreo, requiere 648 Mbytes y una película de 90 minutos ocuparía 120 Gbytes, ambos sin comprimir.

Por ello, la información multimedia se trata casi exclusivamente de forma comprimida. El ancho de banda dependerá por tanto del tipo de compresión y de la calidad con que se quiera transmitir. Los tres estándares más difundidos para compresión de vídeo: *ISO Moving Pictures Expert Group (MPEG)*, *Intel's Digital Video Interactive (DVI)* y *International Telecommunications Union (ITU) H.261*; requieren un ancho de banda de 1.2 a 40 Mbps para MPEG y MPEG-2, 1.2 a 1.8 Mbps para DVI, y de 0.064 a 2 Mbps para H.261 [Stuttgen95]. En el caso de audio, con calidad telefónica el ancho de banda varía de 4 a 32 kbps (dependiendo de la codificación ADPCM o CELP) y con calidad CD y usando compresión MP3 varía de 32 a 256 kbps.

### ***1.2.2.- Retraso de transmisión***

Estos requerimientos son más estrictos que los de ancho de banda. La experiencia con los sistemas de conferencia multimedia y los estándares ITU sugieren un retraso máximo de 150 ms en las aplicaciones de vídeo interactivas. En telefonía este retraso es más exigente (<25ms) para evitar el eco.

Otro tema son los tiempos de compresión y descompresión de las imágenes de vídeo. Siguiendo los requerimientos del CCITT de un máximo de 150 ms de fuente a destino, se pueden identificar las siguientes componentes en el retraso[Stuttgen95]:

1. retraso en la compresión y descomposición en paquetes en la fuente,
2. retraso de transmisión en la red,
3. almacenamiento en el destino y retraso de sincronización,
4. retraso de la composición de los paquetes y la descompresión en el destino.

El vídeo debe tener de 25 a 30 imágenes por segundo. Esto deja un tiempo máximo de compresión-descompresión de 33 a 40 ms ( $30^{-1}$  a  $25^{-1}$ ) para evitar el solapamiento en la compresión de las imágenes. Restando a 150 ms tanto el tiempo de la compresión como el de descompresión deja un retraso máximo de 70 a 84 ms para la transmisión en la red. Asumiendo que una ruta de tres saltos LAN-WAN-LAN es una topología frecuente, y teniendo en cuenta que los elementos de enlace (*gateways*, *routers*, etc.) también contribuyen al retraso, nos queda un retraso máximo aceptable de 10 a 15 ms por salto. Aunque estos cálculos son aproximativos y dependerían de muchos otros factores, nos sirve como una aproximación a los problemas de la transmisión.

Otro parámetro que se suele considerar es la varianza de este retraso (*delay jitter*), es decir, la diferencia entre los retrasos experimentados entre dos paquetes. En determinadas aplicaciones es necesario que los paquetes lleguen con poca variación del retraso, como puede ser el audio. Tener acotada esta varianza del retraso permite calcular la cantidad de memoria necesaria para eliminar esta varianza en la reproducción.

### ***1.2.3.- Fiabilidad***

Las redes tradicionales proporcionan comunicación fiable entre emisor y receptor. Los protocolos de transmisión tienen sistemas de control de errores y de reenvío de paquetes que aseguran que esta fiabilidad es transparente a los niveles superiores.

Para la transmisión en tiempo real esta gestión de errores puede ser negativa, debido al retraso que produciría la retransmisión de un paquete de nuevo. Para evitar este problema se plantea que el tratamiento y gestión de los errores sea a niveles superiores. En una red en tiempo real, la pérdida de paquetes puede ocurrir debido a la saturación de memoria en los nodos (*buffer overrun*) o al superar el retraso máximo exigido.

Las prestaciones de una transmisión multimedia pueden ser medidas en dos dimensiones: latencia y fidelidad. La latencia puede ser vital para aplicaciones interactivas como conferencias mientras que para la transmisión de una película no lo es. La fidelidad de la transmisión es variable. Hay aplicaciones que no toleran ninguna variación en la fidelidad de la imagen como podría ser la transmisión de imágenes médicas y otras en que esta variación sólo produce una cierta distorsión tolerable como la transmisión de películas o música.

#### ***1.2.4.- Sincronización de canales y multidifusión***

Cuando audio, vídeo y otros datos vienen por distintos canales, se necesitan mecanismos para la sincronización de los distintos flujos en el destino. Esto se puede conseguir usando una combinación de asignación de tiempos y almacenamiento antes de su visualización. Esto en general no afecta a la red y es un problema del destino.

La multidifusión (*multicast*) es la habilidad de la red para enviar eficientemente información a múltiples destinos. Esto es una característica común al tráfico multimedia en el que una película puede ser vista por varios espectadores a la vez. Esto permite el ahorro de recursos en la red al utilizar un único canal siempre que sea posible compartirlo por todos los receptores.

### ***1.3.- Calidad de servicio***

En esta tesis se va a usar la siguiente definición de calidad de servicio [Vogel95]:

*“Calidad de servicio representa el conjunto de las características tanto cuantitativas como cualitativas de un sistema distribuido necesarias para alcanzar las funcionalidades requeridas por una aplicación.”*

La funcionalidad incluye tanto la presentación de la información como la satisfacción general del usuario. Normalmente, la calidad de servicio es expresada por medio de parámetros que se negocian. Las necesidades dependerán del tipo de aplicación y pueden variar durante la transmisión.

El primer requerimiento para cualquier transmisión en tiempo real es que todos los niveles que compongan la arquitectura de la red deben garantizar unas prestaciones mínimas y deterministas. En este sentido no se puede utilizar *Ethernet* como protocolo de red aunque usáramos protocolos deterministas a niveles superiores [Banerjea96].

La calidad del servicio vendrá determinado por el punto de vista que se tome. Desde el punto de vista del emisor o receptor los requerimientos están relacionados casi exclusivamente con el tiempo de entrega de los paquetes de información (retraso), la tasa de pérdida de información y el ancho de banda. Otros puntos de vista pueden ser la eficiencia en el uso de la red, la tasa de errores o retransmisiones.

El tráfico se puede dividir en distintas categorías, bien en función de la tolerancia a los parámetros indicados o bien por los requerimientos de los parámetros [Braden94]. En la figura 1.1 el tráfico es clasificado en el producto cartesiano: (sensibilidad al retraso) X (sensibilidad a la pérdida). Como se ve, el grado en que las prestaciones de una aplicación dependen de este retraso varía ampliamente y las podemos catalogar en aplicaciones de tiempo real y aplicaciones elásticas.

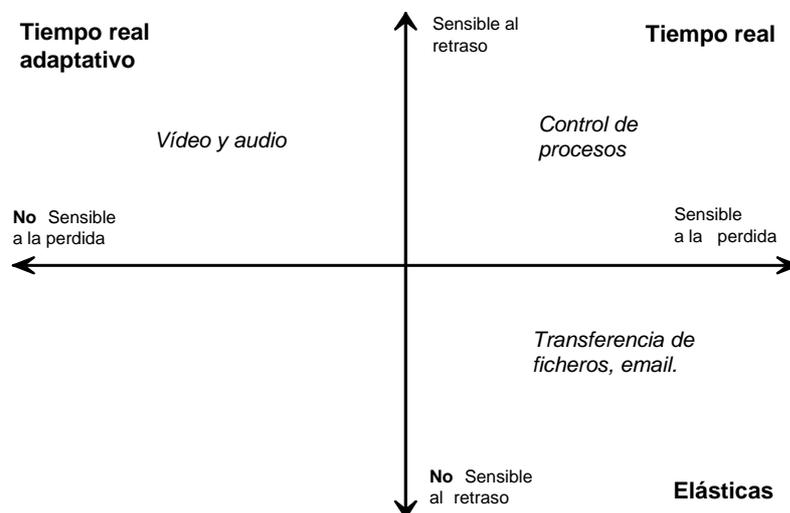


Figura 1.1: Tipos de tráfico en función de la sensibilidad al retraso o pérdida.

### 1.3.1.- Aplicaciones de tiempo real

Una clase importante de estas aplicaciones son las de reproducción. En este tipo de aplicaciones la fuente coge una señal, la convierte en paquetes y los transmite por la red. La red introduce un retraso que debe ser tratado en el receptor. Para poder tratar correctamente los paquetes, la aplicación necesita saber a priori el máximo retraso que los paquetes pueden experimentar.

El retraso puede afectar las prestaciones de las aplicaciones de dos formas. Primero, el tiempo del retraso determina la latencia de la aplicación. Segundo, el retraso individual de los paquetes puede hacer que la fidelidad decaiga si se excede el tiempo de retraso determinado; en este caso la aplicación puede retrasar la ejecución para reproducir estos paquetes retrasados (lo que introduce distorsión) o bien simplemente descartarlos (lo que crea una señal incompleta). En este sentido se pueden distinguir dos tipos de aplicaciones:

- **Aplicaciones intolerantes:** Estas aplicaciones no se pueden adaptar a que un paquete se retrase más del límite predeterminado. Necesitan por lo tanto un límite superior del retraso determinista. Estas aplicaciones requieren un modelo de servicio denominado servicio garantizado o determinista.
- **Aplicaciones adaptativas:** Estas aplicaciones pueden tolerar que lleguen paquetes con un mayor retraso. Estas aplicaciones requieren un modelo de servicio denominado servicio predictivo o estadístico, que proporciona un servicio bastante fiable pero no seguro. Este tipo de aplicaciones pueden aceptar una merma en la calidad del servicio, presumiblemente por el menor coste de este modelo, debido a que se incrementa el uso de los recursos de red.

Para proporcionar un límite en el retraso, el tráfico debe ser caracterizado y la red tiene que ejercer un control de admisión sobre nuevos tráficos que asegure que una petición de flujo puede ser tratada con la calidad de servicio requerida.

### ***1.3.2.- Aplicaciones elásticas***

Estas aplicaciones siempre esperan a que los datos lleguen. Este tipo de aplicaciones no requieren ninguna caracterización del servicio para funcionar. Ejemplos de estas aplicaciones son transferencias (FTP), terminales (Telnet, X-Windows, NFS), etc. Un modelo de servicio para estas aplicaciones es proporcionar un servicio “tan rápido como se pueda” (ASAP: *as-soon-as-possible*). En contraste a los modelos en tiempo real, estas aplicaciones no están sujetas a control de admisión. Suelen estar basadas en un tipo de servicio usualmente denominado “*best-effort*”.

### ***1.3.3.- Ancho de banda y retraso***

Otro aspecto a considerar en el tráfico es el ancho de banda y retraso necesario para la transmisión. En el gráfico de la figura 1.2 el tráfico es clasificado en el producto cartesiano: (ancho de banda) X (retraso). Como se ve, la transmisión de vídeo y audio requiere un gran ancho de banda y bajos retrasos. El caso contrario es el correo electrónico, que tiene bajo ancho de banda y tolera un alto retraso (en la mayoría de los casos).

### ***1.3.4.- Nivel de calidad de servicio***

Cada tipo de aplicación requiere diferentes niveles de calidad de servicio. Bajos niveles de calidad son fáciles de implementar usando mecanismos simples de gestión, como por ejemplo controlando periódicamente el nivel de carga de una red y aumentando

los recursos de la red antes de que estén congestionados. Este esquema podría servir para proporcionar un servicio predictivo.

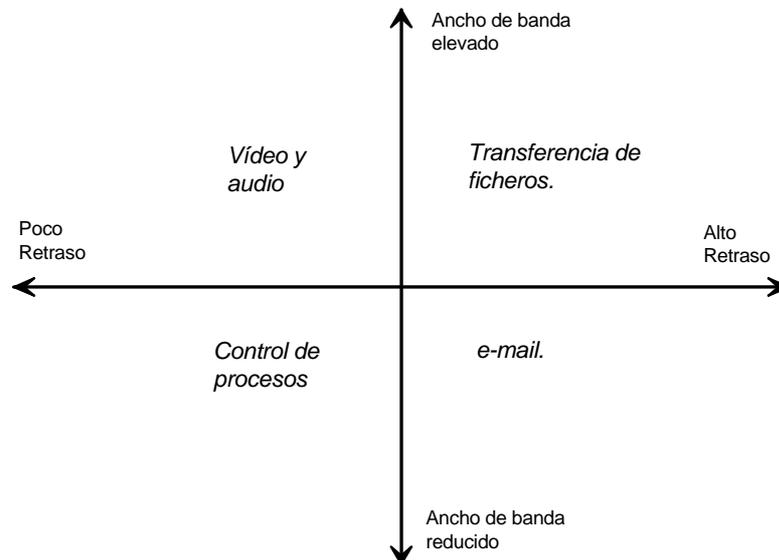


Figura 1.2: Tipos de tráfico en función del ancho de banda y el retraso.

En cambio, obtener una alta calidad de servicio, como pueda ser el servicio garantizado, es más complejo de solucionar. En general, hay que sobredimensionar los recursos de la red y tenerlos asignados para poder garantizar este nivel de servicio. En este sentido, se puede medir la calidad de servicio que ofrece la red en función de la eficiencia de recursos que usa. Este compromiso entre calidad y eficiencia es un aspecto muy importante en el diseño de las redes en tiempo real y es el objeto de estudio principal en esta tesis.

Además, aunque los mecanismos de calidad de servicio pueden variar en distintas partes de la red, esta calidad tiene que obtenerse entre emisor y receptor.

## 1.4.- Modelos de servicios

Un modelo de servicio define las propiedades que debe tener un servicio y que éste ofrece a las aplicaciones que lo usan. En general se puede hablar de dos modelos: servicios integrados (*IntServ*) y servicios diferenciados (*DiffServ*)

### 1.4.1.- Modelo de servicio integrados

El modelo de servicios integrados intenta integrar todos los tipos de tráfico posibles en una misma red de uso general [Braden94]. Este modelo ofrece servicios

cuantificables y mesurables en el sentido que son definidos para proporcionar una determinada calidad de servicio para un tipo de tráfico cuantificado. Este modelo está típicamente asociado a mecanismos de admisión y reserva de recursos en la red.

El modelo de reserva describe cómo una aplicación negocia el nivel de calidad de servicio. El modelo más simple es que una aplicación pida una calidad de servicio particular y que la red se lo proporcione o lo deniegue.

Sin embargo, más que rechazar la petición, la red podría conceder un nivel de recursos menor que el pedido. Un esquema más complejo es el modelo de reserva de “doble pasada”. En este esquema, se propaga la especificación del tráfico inicial desde el origen a los posibles destinos. Cada *router* en las rutas guarda estos valores y quizá los ajusta para reflejar su capacidad disponible. Esta especificación ajustada a la red es devuelta al origen que decide si admite o no el canal.

#### ***1.4.2.- Modelo de servicios diferenciados***

Este modelo es un mecanismo de calidad de servicio de nivel 3 que ha sido utilizado durante algunos años, aunque se ha realizado poco esfuerzo para su estandarización hasta la aparición recientemente del grupo de Servicios Diferenciados de la IETF (DiffServ) [Blake98].

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

De este modo, se pueden tener varias clases de servicio para tiempo real, con varios niveles de retraso. También habrá niveles con servicio predictivo y otros sólo con garantía de entrega. El cliente escogerá el tipo de servicio en función del tráfico a transmitir y por supuesto, el precio que quiera pagar.

Otra de las ventajas de este modelo es su menor complejidad de implementación y su fácil integración con los protocolos IP, en el que cada paquete puede ser marcado con la clase de servicio que requiere. Esta marca será utilizada por los *routers* para diferenciar el servicio por paquete.

#### ***1.4.3.- Tráfico agregado y por conversación***

Es importante determinar cómo se va a gestionar el tráfico internamente en la red. La gestión del tráfico por conversación trata cada conversación como un flujo separado.

Tradicionalmente, este tipo de gestión está asociado al modelo de servicios integrados. En este caso, la red asigna recursos independientes al resto de las conversaciones y mantiene un control de ellos.

En el núcleo de grandes redes, donde es posible soportar cientos de miles de conversaciones simultáneamente, este mecanismo no es práctico. En estos casos se utiliza el tráfico agregado. De esta forma, un conjunto del tráfico de diferentes conversaciones, se clasifica como un mismo flujo y se maneja como un tráfico agregado. Además, esta agregación permite reducir en conjunto los recursos necesarios y permite obtener una calidad de servicio casi equivalente al modelo por conversación. Los servicios diferenciados son claros ejemplos de uso de tráfico agregado.

#### ***1.4.4.- Requerimientos para compartir recursos***

Normalmente la red va a ser compartida por distintos tipos de tráfico. Mientras el aspecto más importante en la calidad de servicio es el retraso, aquí el interés primario es el ancho de banda de los enlaces. Este componente del modelo de servicio, llamado partición de enlaces, contempla como compartir el ancho de banda de un enlace entre varios flujos de acuerdo con ciertas características [Floyd95]. Se pueden distinguir los siguientes modos de compartir:

- **Partición multi-entidad:** Un enlace puede ser usado por varias organizaciones. Se debe asegurar que los enlaces son compartidos de forma controlada, quizás de forma proporcional a lo que paguen.
- **Partición multi-protocolo:** Se debe prevenir que una familia de protocolos (DECnet, IP, IPX, OSI, SNA, etc. ) sobrecargue un enlace y excluya al resto. Esto es importante porque las distintas familias de protocolos responden de forma diferente a la congestión.
- **Partición multi-servicio:** Un administrador de red puede limitar la fracción de ancho de banda para cada clase de servicio.

El control de admisión o la diferenciación de servicios será necesario de nuevo para asegurar que los recursos se compartan de forma adecuada.

## 2.

# Transmisión en tiempo real

### 2.1.- Introducción

En los primeros apartados de este capítulo se describe cuales son los componentes que forman parte de las redes para transmisión en tiempo real. A continuación se describirán tres enfoques distintos de cómo implementar redes para la transmisión en tiempo real.

El objetivo de una transmisión en tiempo real es asegurar una calidad de servicio entre emisor y receptor. Esta transmisión normalmente atraviesa varios tipos de redes como se muestra en la figura 2.1. En esta red se quiere establecer un canal de comunicación entre un ordenador conectado a una red privada a un ordenador conectado a una pública. Este sería el típico caso de comunicación entre un empleado de una empresa y un cliente desde su casa. La red privada de la empresa, que es una red local, está conectada a la red pública 1 y ésta a la red pública 2 donde está conectado el receptor.

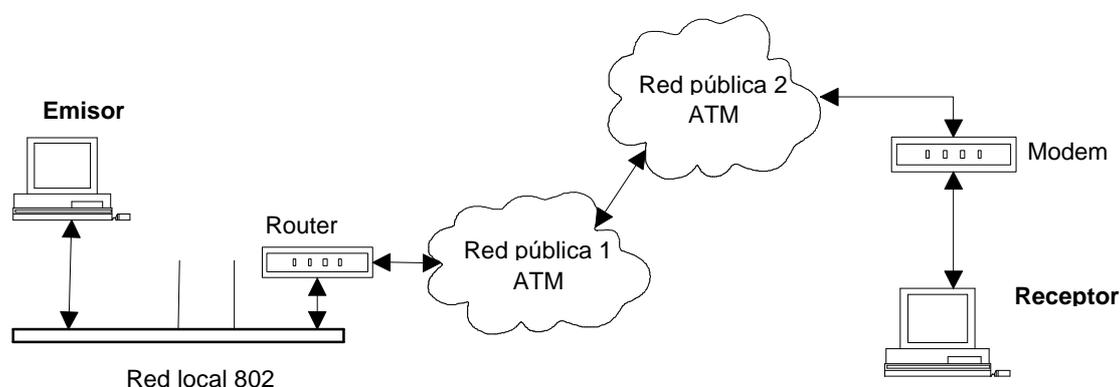


Figura 2.1 Red ejemplo.

La calidad de servicio siempre se va a referir de emisor a receptor. Pero ésta va a depender de la calidad de servicio que pueda ofrecer cada subred. Por ello se necesitan mecanismos globales que gestionen esta calidad y negocien con las subredes la calidad de servicio individualmente. Cada subred tiene que proporcionar mecanismos locales de calidad de servicio como puede ser ATM o 802.1p. Así, la calidad de servicio global será consecuencia de las calidades de servicio negociadas en cada una de las subredes. La gestión de calidad de servicio en cada subred vendrá determinada por el modelo de servicio, que podrá ser de servicios integrados o diferenciados. Este modelo determinará como se gestionan los recursos en la subred y la relación con el resto de las subredes.

Un aspecto importante para gestionar la calidad de servicio es la señalización, es decir, los mensajes que se envían los distintos componentes de la red para gestionar los recursos. Estos mensajes son necesarios para poder garantizar esta calidad de servicio pero al coste de consumir recursos. En este sentido, si se aumenta la señalización entre elementos se puede sobrecargar la red. Una alternativa es que se administre la red para que sólo determinados dispositivos críticos participen en la señalización y el control de admisión. Por ello, es muy importante al diseñar una red en tiempo real, tener en cuenta la sobrecarga que suponen los protocolos que ofrecen calidad de servicio.

Aunque en esta tesis se ha seguido el modelo de servicios integrados, al final de este capítulo se describe brevemente el funcionamiento de una red de servicios diferenciados. Además, se explica como encajar dentro de una red de servicios integrados otra de servicios diferenciados sin demasiadas complicaciones.

## ***2.2.- Componentes comunes para la transmisión en tiempo real***

Los servicios de comunicaciones para la transmisión multimedia tienen en general dos requerimientos: especificar y asegurar la calidad de servicio y ofrecer soporte para la comunicación de grupos. La comunicación de audio o vídeo requiere la provisión de cierto nivel de calidad (QoS), garantizándolo durante el tiempo de la transmisión. Este nivel de calidad lo tienen que establecer las aplicaciones a la hora de comunicarse. Por lo tanto se tienen que proveer mecanismos para :

- el establecimiento y corte de los canales,
- la negociación de los niveles de calidad entre emisor y receptor, sistemas intermedios y control de red, y
- el control del nivel de calidad acordado.

La comunicación multimedia se suele realizar para grupos de más de dos usuarios. Los grupos pueden:

- tener miembros estáticos o dinámicos durante su vida,
- tener control de acceso de los miembros centralizado (por el emisor) o distribuido (por el receptor), y
- consistir en miembros con características y requerimientos homogéneos o heterogéneos.

En general, proporcionar este tipo de servicios es más complejo que implementar una pila de protocolos TCP o OSI. En cambio, cuando se ha establecido la comunicación los requerimientos son menores en conceptos como tratamiento de errores y control de flujo. La implementación de estos sistemas incluye el desarrollo de nuevos *routers* que soporten este tipo de transmisión. Cada *router* sólo debe admitir nuevas comunicaciones si permiten obtener la calidad requerida.

Las redes basadas en paquetes como IP tienen como objetivo maximizar la utilización de la red por medio de la multiplexación de los canales. Además pueden proporcionar comunicación multipunto, y proporcionar robustez adaptándose a la dinámica de la red. Sin embargo, este funcionamiento hace que el comportamiento sea difícilmente predecible. En cambio, las redes basadas en conexión proporcionan un servicio garantizado, pero hacen un uso no eficiente de los recursos de la red, no se adaptan a los fallos de la red y no soportan comunicaciones multipunto.

El desarrollo de este tipo de redes requiere distintos aspectos, que incluyen:

1. **Especificación del flujo:** Una especificación del flujo definiendo el tipo del tráfico, los requerimientos del receptor y la calidad de servicio que se va a proporcionar al flujo.
2. **Encaminamiento:** La red debe decidir como transportar los paquetes desde la fuente al destino. Para ello se requiere un protocolo de encaminamiento que soporte calidad de servicio y rutas *unicast* y *multicast*.
3. **Reserva de recursos:** Para mantener un flujo con una calidad de servicio se requiere un protocolo de reserva para crear y mantener las reservas de recursos, como son el ancho de banda o número de *buffers*.
4. **Control de admisión:** Un algoritmo de control de admisión para mantener la carga de la red a un determinado nivel.

5. **Planificador de paquetes:** Un algoritmo de servicio de paquetes para planificar la transmisión de los paquetes con el objetivo de mantener el servicio garantizado para cada flujo.

### 2.3.- Modelo conceptual de una red en tiempo real

Se puede modelar una red en tiempo real y concretamente los canales, como una red de flujos, en el que cada canal es un flujo [Cobb97]. El emisor introduce en la red un flujo de entrada  $E(t)$  (la información a transmitir) y el receptor recibe un flujo de salida  $R(t)$  (la información recibida) como se muestra en la figura 2.2.

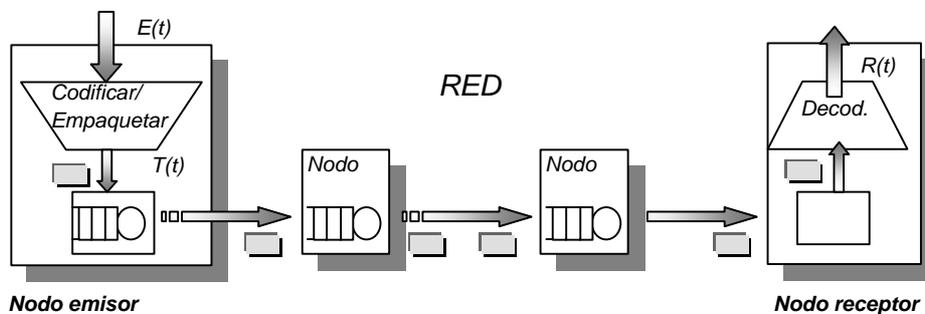


Figura 2.2: Modelo de red en tiempo real.

El emisor introduce el flujo en la red a una determinada tasa. Este flujo se cuantifica en paquetes en el emisor. Cuando un paquete de un flujo llega a un nodo (a través de su enlace de entrada) lo pasa al nodo de salida aplicando una disciplina de servicio: si en ese momento no lo puede transmitir lo almacena en una cola, lo cual provocará un retraso en el flujo. Cuando este paquete llega al receptor se decodifica para formar el flujo de salida  $R(t)$ . Por lo tanto, el almacenamiento del flujo en la red es lo que provoca el retraso de emisor-receptor como se puede ver en la figura 2.3a.

La función  $E(t)$  representa los bits enviados en la red para el tiempo  $t$  y  $R(t)$  los bits recibidos en la red para el mismo tiempo  $t$ . La representación de estas dos funciones (figura 2.3) es útil para calcular dos valores: el retraso en función del tiempo (el valor  $d$ ) y la cantidad de la información almacenada en la red (valor  $b$  o *backlog*). Una forma de calcular una cota del retraso máximo es obtener el máximo valor de  $d$  para cualquier tiempo  $t$ . En el caso de que la red no pueda servir el tráfico introducido a la tasa pedida se producirá un almacenamiento de la red (incremento de  $b$ ) hasta que se produzca pérdida de información (figura 2.3b). Si la red puede admitir el tráfico a la misma tasa que la pedida,

el almacenamiento será 0 y el retraso sólo dependerá de la latencia de la red (retraso constante) como se muestra en la figura 2.3c.

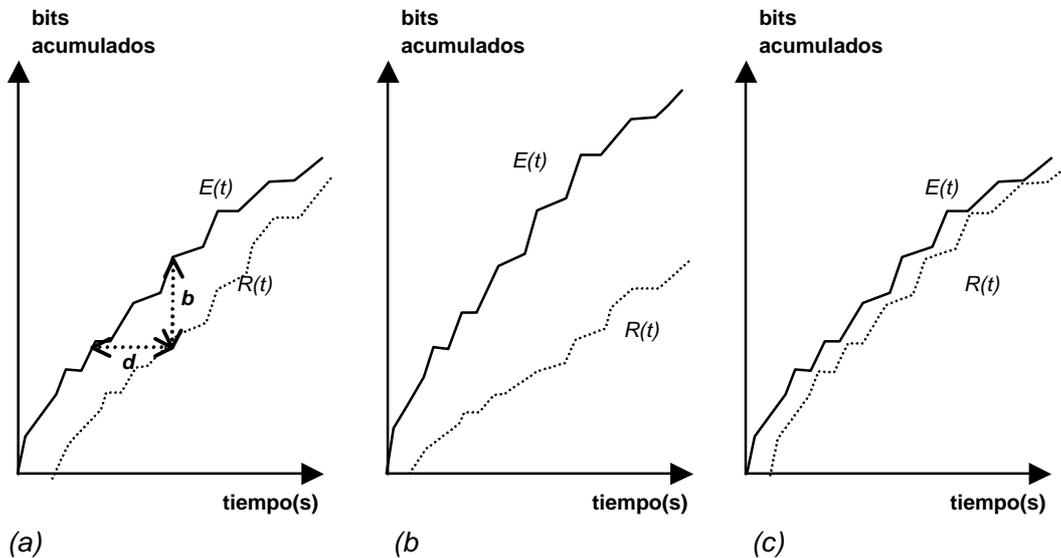


Figura 2.3: Tasa de envío vs. tasa de recepción.

## 2.4.- Especificación y conformación del tráfico

Una especificación de flujo es un acuerdo entre todos los componentes de una red para especificar el tráfico que va a tener de una forma precisa y predeterminada [Tanenbaum96]. Consiste en una serie de parámetros que describen como el tráfico es introducido en la red y la calidad de servicio deseado por las aplicaciones. La idea es que antes de establecer una conexión, el origen del flujo informe sobre las características del flujo a transmitir y el servicio deseado (especificación de la calidad de servicio). Toda esta información es la que compone la especificación del flujo.

Uno de los componentes más importantes de esta especificación es la descripción de cómo se va introducir el tráfico en la red que se suele denominar modelo del tráfico. El objetivo es regular el tráfico a transmitir con el objeto de eliminar la congestión en la red debido a las características de gran variabilidad del tráfico. Este mecanismo de regulación del tráfico de acuerdo al modelo del tráfico se denomina conformación del tráfico (*traffic shaping*).

La conformación de tráfico es un mecanismo de gestión de la congestión en bucle abierto (*open loop*) que permite a la red saber cómo es el tráfico que se transmite para poder decir si lo pueda manejar. Al hecho de monitorizar el tráfico para que cumpla el patrón acordado se denomina comprobación del tráfico (*traffic policing*).

Los modelos de tráfico más comunes son el *leaky bucket* y *token bucket*. Otros esquemas como el D-BIND [Knightly94], *double leaky bucket* o modelos multi-parámetros son ampliaciones de éstos usados con planificadores complejos. También se describe el modelo *Tenet* ( $X_{min}$ ,  $X_{ave}$ ,  $I$ ,  $S_{max}$ ) [Ferrari90a] por ser un planteamiento diferente a los anteriores.

### 2.4.1.- Algoritmo *leaky bucket*

Este algoritmo fue introducido por Turner [Turner86] y desde entonces ha sido el más ampliamente usado para describir tráficos. Este algoritmo regula el tráfico a modo de un cubo con goteo tal como se representa en la figura 2.4. Se usan dos parámetros para describir el algoritmo: la capacidad del cubo  $s$  (bits) y la tasa de drenaje  $r$  (bits/s). El funcionamiento del algoritmo es simple: siempre que el cubo tenga contenido se envía a la red con tasa  $r$ , todo paquete entrante se introduce en el cubo y en el caso de que el cubo esté lleno, el paquete entrante se pierde. De esta forma, se limita la tasa de transmisión del tráfico al valor de  $r$ . El valor de  $s$  para un determinado flujo habrá que calcularlo de tal forma que no se pierdan paquetes.

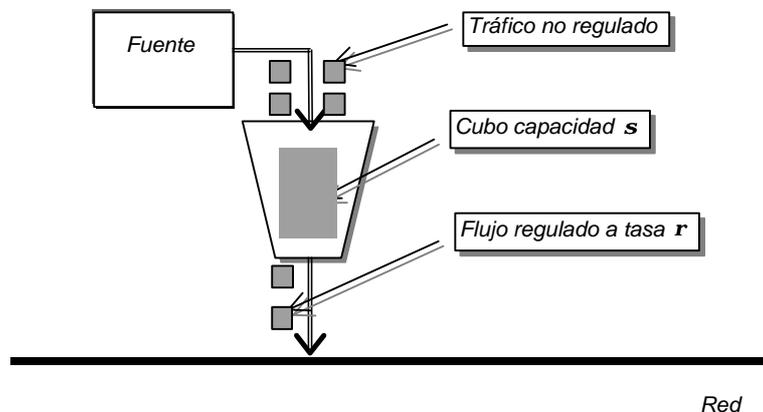


Figura 2.4: Algoritmo *Leaky bucket*.

Para que no se produzca ninguna pérdida de paquetes, el emisor no puede transmitir en el periodo  $[0,t]$  más de  $s+rt$  bits. En general, el valor  $r$  representa la tasa media de transmisión de la fuente.

### 2.4.2.- Algoritmo *token bucket*

El objetivo de este algoritmo es permitir transmitir a mayores velocidades cuando la fuente recibe un pico. El funcionamiento del algoritmo es el siguiente: el cubo contiene *tokens* generados a una tasa  $r$  (véase la figura 2.5). El cubo puede admitir como máximo  $b$

*tokens*, estando al inicio lleno. Para que se transmita un bit se tiene que coger un *token* del cubo y eliminarlo. Mientras existan *tokens* en el cubo, la fuente puede insertar el tráfico a la red a la tasa deseada. Cuando se acaban los *tokens* tendrá que esperar al próximo *token* que se genere, lo que implica que la tasa de transmisión disminuye a  $r$ . En esencia, lo que permite *token bucket* es poder transmitir en un determinado intervalo a tasas superiores a  $r$ .

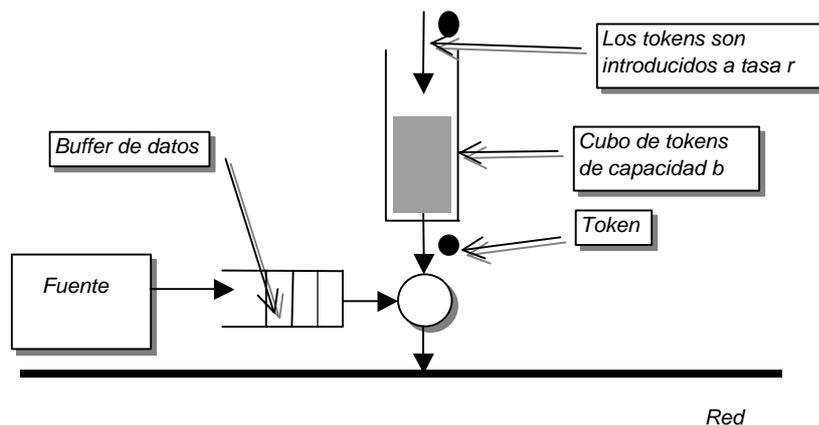


Figura 2.5: Algoritmo token bucket.

El parámetro  $r$  especifica la tasa de datos sostenible continuamente, mientras que  $b$  especifica en cuánto se puede exceder esta tasa para cortos periodos de tiempo. Más específicamente, el tráfico debe obedecer la regla de que para cualquier periodo de tiempo, la cantidad de datos enviados no puede ser superior a  $rt+b$ , para cualquier intervalo de tiempo  $t$ . Además, se suele imponer un límite en la tasa de transmisión que es la tasa pico  $p$ . Con este límite el tráfico no puede exceder  $\min[pt, rt+b]$ .

### 2.4.3.- Modelo Tenet

Este modelo de tráfico fue introducido por Ferrari [Ferrari90a] para regular el tráfico en una red en tiempo real. Un tráfico satisface el modelo  $(X_{min}, X_{ave}, I, S_{max})$  si el tiempo de llegada entre dos paquetes del tráfico es siempre mayor que  $X_{min}$ , el tiempo medio de llegada entre paquetes es  $X_{ave}$  para cualquier intervalo de tiempo  $I$ , y el tamaño máximo de un paquete es menor que  $S_{max}$ .

Con este modelo, el emisor puede enviar a una tasa pico de  $1/X_{min}$  hasta que es forzado a parar la transmisión por el límite impuesto por  $1/X_{ave}$ .

## 2.5.- Planificadores. Disciplinas de servicio

### 2.5.1.- Introducción

Los planificadores de tráfico pueden ser usados en distintos entornos para satisfacer una amplia variedad de objetivos [Varma97]. Una aplicación común de los algoritmos de planificación es proporcionar una calidad de servicio a nivel de red aislando unos tráficos de otros. Los planificadores también pueden ser usados para permitir a los usuarios compartir un enlace de forma equitativa o determinista.

Un planificador puede ser contemplado como un sistema de colas que consiste en un servidor que proporciona servicio a un conjunto de clientes. Los clientes encolan paquetes para ser servidos y estos son escogidos por el planificador basándose en una disciplina de servicio definida por el algoritmo de planificación. La disciplina de servicio puede ser diseñada para cumplir los requerimientos de calidad de servicio deseados por cada cliente.

Los atributos deseables para un algoritmo de planificación son los siguientes [Varma97]:

- **Aislamiento de flujos:** Aislar un canal de los efectos indeseables de otros.
- **Retraso emisor-receptor garantizado:** El planificador debe proporcionar un retraso garantizado de emisor a receptor. Además, es deseable que este límite del retraso dependa sólo de los parámetros de la sesión y que no dependa del resto de las sesiones.
- **Utilización:** El algoritmo debe maximizar el uso de ancho de banda del enlace.
- **Equidad (*Fairness*):** El planificador debe servir a las sesiones con tasas proporcionales a su reserva en cada instante, esto es, distribuyendo el ancho de banda libre proporcionalmente entre las activas. Lo ideal sería que se comportase como un flujo perfecto repartiendo perfectamente el ancho de banda. Pero debido a la cuantificación en paquetes de los flujos esto es prácticamente imposible. Por tanto, se introduce el índice de equidad (WFI: *Worst-case fairness index*) que mide la desviación de servicio ofrecido por un planificador con respecto a un modelo perfecto.
- **Simplicidad de implementación:** El algoritmo de planificación debe ser fácil de implementar y con baja complejidad. Esto es importante si se va implementar por hardware.

- **Escalabilidad:** El algoritmo debe comportarse bien en nodos con un gran número de sesiones y con una variedad de velocidades de enlace.

### 2.5.2.- *Disciplinas de servicio*

El objetivo de los planificadores es asignar los recursos de acuerdo a la reserva realizada con anterioridad con el objetivo de cumplir la calidad de servicio exigida. Tres tipos de recursos son asignados por los planificadores: ancho de banda (*qué* paquete es transmitido), tiempo (*cuándo* es transmitido el paquete) y memoria (*qué* paquetes son descartados), lo que afecta a tres parámetros básicos: rendimiento, retraso y tasa de pérdida.

Aunque las disciplinas de servicio han sido ampliamente estudiadas en el contexto de los sistemas de colas y tiempo real, los resultados de estos estudios no son directamente aplicables en el contexto de la transmisión en tiempo real.

En general, se distinguen dos tipos de disciplinas de servicio en los nodos [Zhang98]:

- ***non work-conserving*** en el que los nodos intentan mantener el modelo del tráfico, aunque esto implique que en determinados periodos no se transmita nada. En este caso, cuando entra un paquete en el nodo se le asocia un tiempo de elegibilidad. En el caso de que no haya paquetes en estado de elegibilidad, no se transmite nada. Ejemplos de estas disciplinas de servicio son las de tasa controlada (*“rate-controlled”*): RCSP [Zhang94] , *Jitter-EDD* [Verma91], *“Stop-and-go”* [Golestani90] y *“Hierarchical Round Robin”* [Kalmanek90]. Cada planificador provoca un retraso acotado y calculable para cada paquete. Dado que cada nodo mantiene el modelo del tráfico el cálculo del retraso total es la suma de los retrasos en cada nodo.
- ***work-conserving*** en el que si existen paquetes en el nodo por transmitir se envían. A este grupo pertenecen *Virtual Clock* [ZhangL90], *Weighted Fair Queuing* (WFQ) y *GPS (General Processor Sharing)* [Demers89] [Parekh92]. Para todos estos esquemas existen funciones para calcular el retraso máximo emisor-receptor que están basadas en el trabajo de Parekh y Gallaguer [Parekh93]. Normalmente, el cálculo del retraso es dependiente de la reserva de ancho de banda en los nodos.

Aunque en esta tesis no se va a tener en cuenta, hay que determinar el coste computacional de los algoritmos de planificación para su implementación en redes de alta

velocidad. Por ejemplo, un planificador FCFS tiene un coste de implementación bajo, pero sólo puede soportar un límite de retraso para todas las conexiones. En el otro extremo, el algoritmo EDD es complejo ya que involucra una operación de búsqueda del paquete con el *deadline* más corto.

Otras disciplinas de servicio gestionan la compartición del enlace de una forma controlada, permitiendo una estructura jerárquica, como el planificador CBQ (*Class-based queueing*) [Floyd95].

A continuación se describen dos disciplinas de servicio que el autor considera los más representativos para cada tipo de servicio: el servicio RCSP del tipo *non-work-conserving* y el planificador WFQ del tipo *work-conserving*.

### 2.5.3.- Servicio RCSP

La disciplina de servicio RCSP (*Rate-Controlled Static Priority*) fue introducida por H. Zhang en el grupo *Tenet* [Zhang94]. Como se muestra en la figura 2.6 un servidor RCSP está formado por dos componentes: un controlador de tasa y un planificador con prioridades estáticas. Conceptualmente, el controlador de tasa está formado por el conjunto de reguladores asociados a cada canal que atraviesa el nodo. Cada regulador es un conformador de tráfico que regula el tráfico de entrada al nodo al modelo de tráfico deseado para el planificador.

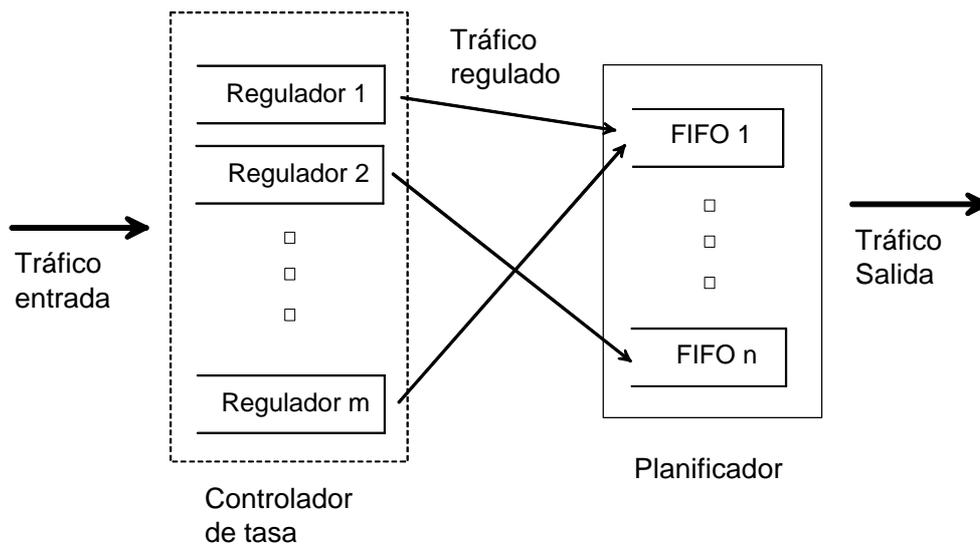


Figura 2.6: Planificador RCSP.

Cuando un paquete llega, el regulador calcula un tiempo de elegibilidad y es retenido en el regulador hasta que cumpla este tiempo. A continuación, se introduce en el planificador deseado en función del nivel asignado. El planificador dispone de un conjunto de colas para cada nivel de prioridad y selecciona los paquetes de la cola más prioritaria que no esté vacía. Cada conexión tiene asignado un nivel de prioridad desde el momento del establecimiento del canal y depende principalmente del retraso exigido.

La forma de calcular el tiempo de elegibilidad depende del modelo del tráfico, el cual va a definir como se regula el tráfico. Para RCSP se utiliza el modelo *Tenet* ( $X_{min}$ ,  $X_{ave}$ ,  $I$ ,  $S_{max}$ ). Con este modelo se obtienen las ecuaciones que definen el tiempo de elegibilidad de un paquete para un canal de tal forma que el tráfico mantiene sus características a lo largo de la red. El tiempo de elegibilidad para el paquete de orden  $k$  de la conexión  $j$  en el nodo  $i$ ,  $e_{i,j}^k$ , se define usando el tiempo de elegibilidad calculado para paquetes anteriores de la misma conexión:

$$e_{i,j}^k = -I, \quad k \leq 0$$

$$e_{i,j}^1 = a_{i,j}^1$$

$$e_{i,j}^k = \max(e_{i,j}^{k-1} + X_{\min}, e_{i,j}^{k-1} \left\lceil \frac{I}{X_{ave}} \right\rceil + I, a_{i,j}^k), \quad k > 1$$

donde  $a_{i,j}^k$  es el tiempo de llegada del paquete  $k$  de la conexión  $j$  al nodo  $i$ .

#### 2.5.4.- Planificador WFQ

En este punto se describe el algoritmo de planificación más utilizado actualmente y que es la base de muchos otros. Este planificador fue introducido con el nombre de *Weighted Fair Queueing* (WFQ) en [Demers89] aunque luego se ha rebautizado como PGPS (*Packet General Processor Sharing*). Es un intento de aproximarse a un modelo de flujo perfecto (denominado GPS: *General Processor Sharing*) con un procesador compartido entre los distintos flujos de acuerdo a unos pesos predeterminados. De esta forma, el ancho de banda se reparte proporcionalmente entre los distintos flujos.

En los algoritmos WFQ, cuando llega una trama se calcula y asocia una etiqueta (*Time Stamp*, TS) que va a determinar el orden de salida y se envía a la cola de su conexión. La siguiente trama a transmitir será la que tenga el valor TS más pequeño.

TS se calcula de acuerdo a siguiente la fórmula:

$$TS_1^0 = 0$$

$$TS_i^k = \max(TS_i^{k-1}, v(t)) + \frac{L_i^k}{r_i}$$

donde:

$v(t)$  es la función de tiempo virtual calculada a la llegada del paquete.

$TS_i^{k-1}$  corresponde al *time stamp* de la trama anterior.

$L_i^k$  es el tamaño de la trama en bits.

$r_i$  es el ancho de banda deseado para la sesión  $i$ .

La función  $v(t)$  se calcula así: si  $B(t)$  representa el conjunto de sesiones con algún paquete en cola (*backlogged*) en el planificador en el instante  $t$  y  $V$  el número total de sesiones, entonces:

$$v(t) = \frac{\sum_{j=1}^V f_j}{\sum_{i \in B(t)} f_i}$$

Donde  $f$  es un número real que indica la porción de ancho de banda del enlace requerido por la sesión que cumple la siguiente condición:

$$r_i \leq \frac{f_i}{\sum_{j=1}^V f_j} r$$

donde  $r$  es la capacidad de enlace. Por ejemplo, se puede usar  $f_i = r_i/r$ .

Cuando una sesión tiene tráfico pendiente, se cumple que  $v(t) \neq TS_i^{k-1}$  por lo que  $v(t)$  no influye en  $TS$ . En caso contrario, al recibir la primera trama después de un periodo sin tráfico, se tiene en cuenta  $v(t)$  para actualizar la sesión. La dificultad del planificador WFQ reside en calcular  $v(t)$ , por lo que se han propuesto otros algoritmos que simplifican su cálculo.

## 2.6.- Gestión del tráfico: control de admisión

Dado que las redes tienen capacidad finita, si no se limita la cantidad de tráfico a transmitir en la red, el servicio ofrecido se degradará, y al final se colapsará. Si la red no puede aceptar un determinado tráfico porque no lo puede gestionar o afectase al resto de tráficos no debería admitirlo.

Las soluciones a este problema se pueden clasificar en dos grupos: reactivas (o esquema de control) y proactivas o (control de admisión) [Zhang98]. Las soluciones reactivas detectan y reaccionan dinámicamente a la congestión dentro de la red, reduciendo o eliminando parte del tráfico. Con este esquema es difícil asegurar la calidad de servicio. En las soluciones proactivas, una conexión se acepta sólo si existen suficientes recursos para satisfacer los requerimientos del nuevo canal y de los ya existentes.

Para ello, con los modelos del tráfico y disciplinas de servicio se han desarrollado unos tests que indican si se puede admitir un nuevo flujo con tales características.

Estos tests pueden ser a nivel de nodo o a nivel de red. A nivel de nodo se chequea con la información disponible del nodo y del flujo si el canal se puede admitir. En el caso de que no se pueda admitir se rechaza directamente la conexión. A nivel de red, el receptor comprueba, con toda la información de los nodos por los que ha pasado el mensaje de conexión, si se cumplen los requerimientos pedidos. En el caso de ser así, se envía al emisor un mensaje de vuelta de establecimiento de canal. En caso contrario, se rechaza el canal y se envía un mensaje de rechazo.

La reserva puede ser en una pasada, en la que los recursos se van reservando en el mensaje de establecimiento de canal, o bien de doble pasada, en el que el mensaje pasa primero por todos los nodos recogiendo información sobre los recursos disponibles. Cuando el mensaje llega al receptor, éste decide si se puede admitir el canal y envía el mensaje de vuelta reservando los recursos en la red.

Los tests van a depender del modelo de los nodos y del tráfico. Cuando más exactos sean los tests se hará un uso más eficiente de los recursos de la red y se podrán admitir más canales.

## 2.7.- Protocolos de comunicación con calidad de servicio

Se pueden distinguir tres niveles de protocolos para ofrecer calidad de servicio [Vogel95][Stuttgen95]:

- Protocolos de bajo nivel. ATM es quizá el protocolo más adecuado para la transmisión multimedia por su gestión de la calidad de servicio. Otros protocolos de bajo nivel pueden ser usados con ciertas restricciones como FDDI o Iso-Ethernet [Ross96].
- Protocolos de red y transporte. El protocolo IPv4 no es apropiado para la transmisión en tiempo real, aunque IPv6 se ha diseñado para poder soportar este tipo de tráfico. Otros esquemas, como la *Suite Tenet*, incluyen nuevos niveles de red y transporte con el objetivo de soportar tráfico en tiempo real.
- Protocolos de aplicación. En este nivel se han desarrollado distintos protocolos como RSVP (*Resource Reservation Protocol*) que gestiona el control de admisión y la reserva de recursos. Otros protocolos se encargan de la transmisión y sincronismo de audio y vídeo como RTSP (*Real-time Streaming Protocol*) o RTP (*Real-time protocol*)

En [Hernandez98a] se ofrece una comparativa de la idoneidad de los protocolos para la transmisión en tiempo real. De este trabajo se extrae la tabla 2.1 que es un resumen de las características de las distintas redes evaluadas.

De la comparativa realizada se saca la conclusión de que existen algunas redes que prometen un buen soporte al tráfico multimedia, pero sólo unos pocas están disponibles en realidad. ATM se configura como la red multimedia por excelencia tanto para WAN como LAN. La diferenciación entre LAN y WAN es cada vez menor y el usuario requiere servicios de red independientes de la organización física y lógica. De hecho, las nuevas redes como ATM asumen este concepto.

A continuación se describen tres protocolos, uno representativo de cada nivel. No es el objetivo describir el funcionamiento de estas redes, sino presentar cuáles son las soluciones aportadas en cada protocolo para la transmisión en tiempo real. ATM es una red que está disponible y de la que se describe principalmente la especificación del tráfico y el control de admisión. La *Suite Tenet* se creó con el objetivo de investigar la transmisión en tiempo real y no se ha llegado a implantar comercialmente. RSVP es principalmente un protocolo de reserva de recursos que puede ser utilizado sobre y con otros protocolos. En este momento hay varios fabricantes que han anunciado su implementación.

Red	Ancho banda (Mbps)	Retraso transmisión	Variación Del Retraso	Multidifusión	Disponible
<i>Ethernet</i>	10	Aleatorio	$\infty$	Sí	Sí
<i>Iso-Ethernet*</i>	10+6	Fijo < 1ms.	0	No	No
<i>Token Ring</i>	4/16	Configuración < 20ms.	$\Delta$	Sí	Sí
<i>100Base-T</i>	100	Aleatorio	$\infty$	Sí	Sí
<i>802.12</i>	100	Configuración < 10ms.	$\Delta$	Sí	No
<i>FDDI</i>	2 x 100	Configuración	$\Delta$	Sí	Sí
<i>FDDI-II*</i>	N x 6	Fijo < 1ms.	0	Sí	No
<i>X-25</i>	<2	Aleatorio	$\infty$	No	Sí
<i>Frame Relay</i>	< 50	Aleatorio	$\infty$	No	Sí
<i>RDSI</i>	N x 0.064	Fijo < 10ms	0	No	Sí
<i>ATM</i>	25 – 155	Fijo < 10ms	$\Delta$	(SÍ)	Sí

\* : Se refiere a los canales isócronos.

**Fijo:** El retraso es fijo para un canal.

**Configuración:** El retraso depende de la configuración de la red (tamaño de los paquetes, número de estaciones, etc. ). Se da un valor para una red configurada para que pueda asegurar una transmisión aceptable de información multimedia.

$\infty$  = Red asíncrona sin control del retraso.

$\Delta$  = Red síncrona con una variación del retraso entre 0 y un valor máximo.

**0** = Red isócrona.

Tabla 2.1: Idoneidad de las redes para transmisión en tiempo real.

## 2.8.- Calidad de servicio en ATM

### 2.8.1.- Introducción

A mediados de la década de los 80 el CCITT comenzó a trabajar en una segunda generación de la RDSI, conocida como RDSI de Banda Ancha. En su reunión de 1988 en Seúl, se propone la recomendación de utilizar la tecnología ATM para la RDSI-BA [Huidobro99].

La tecnología ATM es una arquitectura de conmutación de celdas que utiliza la multiplexación por división en el tiempo asíncrona. Las celdas son las unidades de transferencia de información en ATM y se caracterizan por tener un tamaño fijo de 53 bytes. Esto permite simplificar los nodos y que la conmutación sea realizada por hardware, consiguiendo con ello alcanzar altas velocidades.

El soporte de múltiples servicios con diferentes calidades de servicio (QoS) y características del tráfico, requiere una extensa y compleja infraestructura de red. La tecnología de red ATM estandariza un conjunto completo de funciones para la gestión del tráfico que pueden ser implementados con varios niveles de complejidad para obtener diferentes niveles de eficiencia [Giroux99]. Ya que esta estandarización es seguida por el

Forum ATM y la ITU-T, las redes ATM pueden interoperar con los equipos de distintos vendedores.

La red define tres niveles:

- **Nivel físico.** Se han definido distintos medios de transmisión como fibra óptica a 155-Mbps, o 100-Mbps FDDI para ATM de área local, además de otras opciones. En el futuro, los interfaces a ATM evolucionarán hasta 622 Mbps o incluso 2.4 Gbps.
- **Nivel ATM.** El nivel ATM es una capa independiente para la conmutación y el multiplexado de los paquetes. Se define la estructura de las celdas como contenedores de información de 53 octetos ( 5 octetos para cabecera más 48 de datos). El encaminado se basa en los identificadores de circuito virtual (VCI: *virtual circuit identifiers*) y los identificadores de camino virtual (VPI: *virtual path identifiers*). Además, ATM es una red orientada a la conexión. Debido al pequeño tamaño de las celdas ATM, la alta velocidad de los enlaces y la velocidad de conmutación de los nodos, ATM proporciona unas latencias mínimas.
- **Nivel de adaptación ATM (AAL).** Este nivel está diseñado como puente entre el nivel ATM y de aplicación.

### 2.8.2.- Componentes de la gestión del tráfico

Una aplicación negocia el **(1) contrato del tráfico** con la red para cada conexión virtual. Este contrato es un acuerdo entre el comportamiento del tráfico que se va a transmitir y el nivel de servicio que se requiere para la conexión. Cuando el tráfico está definido, la red aplica un **(2) control de admisión de la conexión** (CAC) para decidir si se puede admitir con la calidad de servicio exigida con los recursos disponibles en la red. Si la conexión es aceptada, la transmisión puede comenzar. Para garantizar la calidad de servicio durante la transmisión, la red tiene que asegurar que el tráfico de la conexión cumple el contrato. Para ello, la red puede aplicar **(3) funciones de conformidad** para asegurar que un tráfico que no cumpla el contrato no afecte al resto de las conexiones. El tráfico puede ser encolado antes de ser transmitido. Las colas son servidas de acuerdo a **(4) algoritmos de planificación** específicamente diseñados para cumplir la calidad de servicio de los distintos tipos de conexiones. Aunque se ha realizado un control de admisión puede ocurrir congestión en la red. El **(5) control de la congestión** gestiona que se eliminen las celdas de una manera no traumática.

### 2.8.3.- *El contrato del tráfico*

Una aplicación que use una red ATM acuerda sus requerimientos de ancho de banda y prestaciones por medio de un contrato del tráfico. Este contrato incluye los siguientes componentes:

- la categoría del servicio,
- la calidad de servicio requerida,
- las características del tráfico de la conexión,
- la definición de cómo el tráfico se debe comportar (modelo del tráfico).

### 2.8.4.- *Categorías de servicio*

Se definen 5 categorías de servicio (GFR no se describe por estar bajo discusión todavía):

#### 2.8.4.1.- *Servicio CBR*

El servicio CBR (*Constant Bit Rate*) está diseñado para soportar aplicaciones en tiempo real. Proporciona una conexión con ancho de banda dedicado, con ínfima probabilidad de pérdida de celdas, así como retraso bajo y predecible. El tiempo entre celdas es constante y está caracterizado por una tasa pico (PCR: *Peak Cell Rate*).

#### 2.8.4.2.- *Servicio VBR*

El objetivo del servicio VBR (*Variable Bit Rate*) es hacer más eficiente el soporte de aplicaciones de vídeo o otras aplicaciones con tráfico a ráfagas. El tráfico está caracterizado por una tasa de tráfico sostenida (SCR: *Sustained Cell Rate*) y la tasa pico (PCR). El valor SCR es medido sobre un periodo específico y representa la tasa de transmisión media. Hay dos subcategorías de servicio: rt-VBR y nrt-VBR. El tráfico rt-VBR tiene requerimientos estrictos de retraso. El tráfico nrt-VBR no tiene estos requisitos y se puede almacenar en la red (*buffering*).

#### 2.8.4.3.- *Servicio ABR*

El objetivo del servicio ABR (*Available Bit Rate*) es que los nodos terminales participen activamente en la gestión del tráfico. La red informa de cual es su estado y

capacidad y los nodos terminales transmiten de acuerdo a esta información intentando evitar pérdidas de celdas.

#### 2.8.4.4.- Servicio UBR

Las conexiones UBR (*Unspecified Bit Rate*) comparten el ancho de banda no usado por el resto de tipos de servicio, sin ningún tipo de *feedback* por parte de la red. De esta forma la aplicación accede a todo el ancho de banda que la red pueda proporcionar pero sin ninguna garantía acerca de la tasa de pérdida de celdas. Es un servicio tipo “*best-effort*”.

Los tipos de servicio más interesantes para transmisión en tiempo real son CBR y rt-VBR.

#### 2.8.5.- Parámetros de calidad de servicio

La calidad de servicio en ATM está definida por un conjunto de parámetros requeridos para la conexión entre emisor y receptor. Los siguientes tres parámetros son negociados entre la aplicación y la red:

- *cell loss ratio* (CLR),
- *maximum cell transfer delay* (Max-CTD), y
- *peak-to-peak cell delay variation* (P2P-CDV).

y otro tres son impuestos por la red:

- *cell error ratio* (CER),
- *severely errored cell block ratio* (SECBR), y
- *cell misinsertion rate* (CMR).

Se van a describir únicamente los parámetros de retraso. El *cell transfer delay* se define como el tiempo transcurrido entre el envío de una celda por parte del emisor hasta su recepción en destino. Este retraso es la acumulación de los retrasos acumulados en cada nodo y enlace de la red.

**Max-CTD** representa el  $(1 - \alpha)$  cuantil de la función de densidad del CTD, donde las celdas que exceden el máximo retraso son consideradas perdidas (véase la figura 2.7). El ajuste de  $\alpha$  lo realiza la red, pero se escoge para que no supere el valor CLR (*cell loss ratio*).

**P2P-CDV** representa la diferencia entre el máximo y mínimo retraso que una celda puede experimentar.

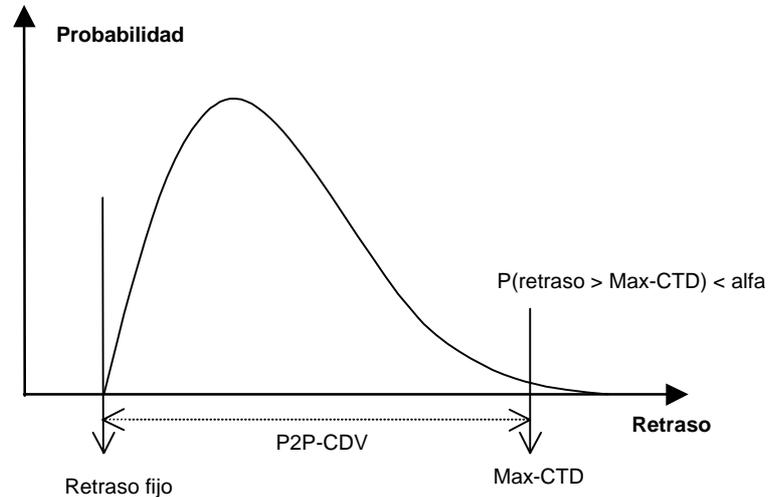


Figura 2.7: Densidad de la probabilidad del retraso.

### 2.8.6.- Modelo del tráfico

Una conexión ATM caracteriza el tráfico usando los siguientes parámetros:

- *peak cell rate* (PCR),
- *sustainable cell rate* (SCR) y *maximum burst size* (MBS),
- *minimum cell rate* (MCR), y
- *maximum frame size* (MFS).

Cada conexión tiene dos conjuntos de descriptores para el tráfico de entrada y salida.

**PCR** representa la tasa pico de emisión de la fuente. Su inversa representa el tiempo mínimo entre paquetes. La unidad de PCR es celdas/segundo. La figura 2.8 representa un PCR contratado de 1/3 de la tasa de la línea.



Figura 2.8: Tráfico CBR.

**SCR** es un límite superior de la tasa media de transmisión de una conexión ATM. La inversa de SCR representa un límite superior del tiempo entre celdas a largo plazo con respecto a la velocidad del enlace. El valor SCR se especifica siempre junto al valor MBS. El valor **MBS** representa el factor de ráfagas (*burstiness*) de la conexión. Especifica el máximo número de celdas que pueden ser transmitidas por una fuente a la tasa pico (PCR) cumpliendo el valor SCR negociado. La figura 2.9 representa una transmisión con un PCR =  $\frac{1}{2}$  de la tasa de enlace y SCR =  $\frac{1}{4}$  con un MBS de 5 celdas. La conexión puede enviar como máximo cinco celdas en diez unidades de *slot*. Alternativamente puede enviar una celda cada 4 pero sin aprovechar la capacidad de emitir en ráfagas.

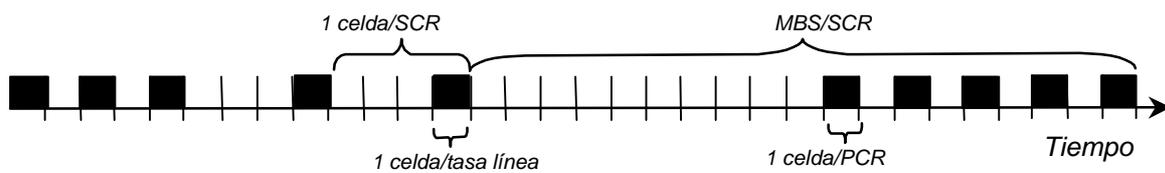


Figura 2.9: Tráfico VBR.

Los parámetros MCR y MFS son usados en los tráficos GFR y ABR. La tabla 2.2. define los parámetros necesarios para cada tipo de servicio [ForumATM96].

Atributos	CBR	RT-VBR	NRT-VBR	ABR	UBR
PCR	✓	✓	✓	✓	✓
SCR,MBS		✓	✓		
MCR				✓ (Opcional)	
CLR	✓	✓	✓		
Max-CTD	✓	✓			
P2P-CDV	✓	✓			

Tabla 2.2: Parámetros para cada tipo de servicio.

### 2.8.7.- Ajuste del tráfico

La reserva de recursos en la red no es suficiente para asegurar la calidad de servicio. Si una conexión excede el tráfico contratado puede incidir en su calidad de servicio y lo que es peor en la calidad de servicio de otras conexiones. Para evitar esto, la red tiene que asegurar que el tráfico se ajuste a los parámetros contratados. Esta capacidad se obtiene por la implementación de los mecanismos de conformación y comprobación del tráfico (*traffic-shaping*, *traffic-policing*, y *soft-policing*).

Para ello se utiliza un algoritmo denominado *Generic Cell Rate Algorithm* (GCRA). Cuando una celda no se ajusta al tráfico especificado puede ser degradada (baja su clase), descartada o aceptada (si no influye al resto de las conexiones). Para evitar la pérdida de celdas, la fuente de tráfico realiza un conformado del tráfico, lo que implica el retraso de ciertas celdas hasta que se ajusten a las especificaciones. El algoritmo GCRA puede ser implementado por medio de un algoritmo *leaky bucket* o como un algoritmo *virtual-scheduling*.

### 2.8.8.- Control de admisión de conexiones

Una conexión ATM atraviesa un conjunto de nodos en la red. Para verificar la admisión de una conexión, el CAC (*connection admission control*) sigue el siguiente procedimiento en cada punto de la red:

1. Mapea la descripción del tráfico asociada con la conexión en un modelo de tráfico.
2. Usa este modelo de tráfico con un modelo apropiado de la cola del nodo para estimar si hay suficientes recursos para admitir la nueva conexión con la calidad de servicio pedida.
3. Reserva los recursos si son suficientes y admite la conexión.

Dependiendo de los modelos de tráfico usados, los procedimientos del CAC pueden ser demasiado conservativos y reservar demasiados recursos. Esto reduce la ganancia estadística que se pueda obtener. La ganancia estadística se define como:

$$\text{GananciaEstadística} = \frac{N^{\circ} \text{ de conexiones admitidas con multiplexación estadística}}{N^{\circ} \text{ de conexiones admitidas con tasa pico}}$$

Dado que las conexiones no envían continuamente datos, es posible reservar menos recursos cuando las conexiones son multiplexadas en un punto. Esto significa que se pueden admitir más conexiones que si se usa el valor PCR en cada conexión.

Un CAC eficiente debe intentar obtener la máxima ganancia estadística posible sin afectar la calidad de servicio. Las categorías de servicio CBR y rt-VBR tienen requerimientos de retraso. Esto es, el retraso en un nodo está acotado, lo que fuerza que el tamaño de los *buffers* sean pequeños y la congestión de celdas prevalece para estos servicios. En consecuencia es difícil obtener multiplexación estadística para estas clases.

Normalmente el CAC para el CBR podría utilizar la siguiente ecuación:

$$\sum_i PCR_i \leq \text{Capacidad del enlace}$$

Este algoritmo CAC está basado en la tasa pico, pero dado que el tráfico no conserva las características se podrían perder celdas. En cambio, para el tráfico VBR la tasa SCR/PCR define el factor de ráfagas y tiene un gran impacto en la ganancia estadística. Si este ratio es bastante menor que uno puede resultar en una utilización pobre del enlace. Por tanto, el CAC admite conexiones si:

$$\sum_i a_i \leq \text{Capacidad del enlace}$$

donde  $SCR_i \leq a_i \leq PCR_i$  para cada conexión  $i$ . La ganancia estadística puede ser definida como el ratio  $\Sigma PCR_i / \Sigma a_i$ . El valor  $a_i$  se llama ancho de banda efectivo o ancho de banda equivalente. Este valor depende de las propiedades estadísticas de la conexión a admitir así como de las colas de la red. En general, para una conexión es intuitivo pensar que el ancho de banda efectivo estará más cerca del valor PCR cuando el tamaño de las colas es reducido y más cerca de SCR cuando las colas son mayores.

Dos propiedades constituyen las principales ventajas de este método:

1. Propiedad de adición: El ancho de banda efectivo total necesario para  $N$  conexiones es igual a la suma de los ancho de banda efectivos de cada conexión.
2. Propiedad de independencia: El ancho de banda efectivo para una conexión dada es sólo una función de los parámetros de la conexión.

Estas dos propiedades hacen que este método sea ampliamente usado en las tecnologías ATM. Existen varios métodos para calcular el valor de  $a_i$  como el *Rate Envelope Multiplexing* (para tamaños de colas pequeños) y *Rate Sharing* [Kelly91].

Aparte de estos métodos analíticos de control admisión, existen también métodos dinámicos que se basan en el estado real de la red para determinar la admisión de un nuevo canal.

## 2.9.- Tenet suite

### 2.9.1.- Introducción

El *Tenet Group* de la *University of California at Berkeley* ha diseñado, simulado e implementado un conjunto de protocolos para soportar canales en tiempo real [Banerjea96][Ferrari92a]. La propuesta intenta ser una solución completa y está más orientado a la investigación y solución de los problemas existentes en este tipo de redes. Los protocolos *Tenet* están diseñados para coexistir con los protocolos Internet.

Los canales en tiempo real son establecidos en la fase de conexión que precede a la transferencia de datos. Un mensaje se envía desde el origen del canal y viaja al destino, provocando que en cada nodo se ejecuten unos tests de admisión para comprobar si se puede obtener la calidad de servicio requerida. Cuando llega el mensaje al destino, éste envía un mensaje de aceptación del canal que llegará al origen estableciéndose el canal.

El grupo *Tenet* tiene también modelos matemáticos y simulaciones de los protocolos utilizados [Ferrari92a][Ferrari92b][Ferrari91][Widyono94].

### 2.9.2.- El diseño de los protocolos Tenet

La *suite Tenet* está formada por la pila de protocolos descritos en la figura 2.10 y tabla 2.3. El módulo RCAP (*Real-time Channel Administration Protocol*) proporciona los servicios de control. Su función principal son el establecimiento y liberación de los canales en tiempo real.

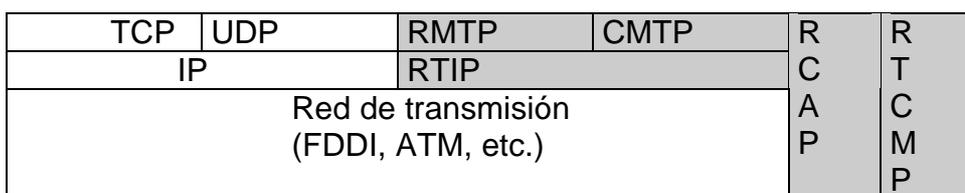


Figura 2.10: Pila de protocolos de Tenet Suite.

Módulo		Descripción
RMTP	Real time Message Transport Protocol	Protocolo de transporte orientado a mensaje.
CMTp	Continuos Media Transport Protocol	Protocolo de transporte orientado a flujo.
RTIP	Real time internet protocol	Protocolo de red en tiempo real.
RCAP	Real time Channel Administration Protocol	Protocolo de establecimiento y control de canales.
RTCMP	Real time Control Message Protocol	No implementado.

Tabla 2.3: Descripción de los módulos del protocolo Tenet Suite.

El establecimiento de un canal se realiza en una sola pasada. Un mensaje `establish_request` se envía desde el origen al destino del canal. Cada entidad RCAP mantiene una tabla de rutas para calcular el próximo salto para el establecimiento del canal. Si el canal puede ser soportado por este nodo (mediante la realización de un test de control de admisión) se reservan de forma provisional los recursos necesarios y el mensaje se envía al siguiente nodo. Si el nodo determina que no puede soportar este canal se envía un mensaje `establish_denied` de vuelta al origen. Este mensaje libera los recursos reservados en cada nodo cuando pasa de vuelta. Cuando el mensaje `establish_request` alcanza el destino, éste toma la última decisión de aceptar o rechazar el canal. Si se acepta el canal, se envía de vuelta al origen el mensaje `establish_accept`. Cuando este mensaje llega a cada nodo intermedio, el RCAP local puede reducir la asignación de recursos realizada en la ida del mensaje `establish_request`. Cuando la asignación definitiva de recursos se realiza, se informa al agente RTIP que se ha establecido un nuevo canal y se envía el mensaje al nodo siguiente de vuelta. Cuando este mensaje llega al origen, la transmisión sobre el canal puede comenzar.

Además, existen mensajes para consultar el estado de un canal en un nodo y para cerrar un canal tanto desde el emisor como el receptor. Los mensajes del RCAP están descritos en la tabla 2.4.

Path message	Sentido	Descripción
<code>Establish_request</code>	Ida	Petición de establecer un nuevo canal.
<code>Establish_accept</code>	Vuelta	Aceptación de un canal.
<code>Establish_denied</code>	Vuelta	Se ha rechazado el establecimiento de un canal.
<code>Status_request</code>	Ida	Petición del estado de un canal en un nodo.
<code>Status_report</code>	Vuelta	Retorna los datos recolectados por un mensaje <code>status_request</code> .
<code>Close_request_forward</code>	Ida	Cierra un canal a petición del origen.
<code>Close_request_reverse</code>	Vuelta	Cierra un canal a petición del destino.

Tabla 2.4: Mensajes de Tenet.

### 2.9.3.- Especificación del tráfico

El cliente debe establecer los requerimientos de calidad de servicio y una descripción del tráfico que va a transmitir en la red. Los parámetros que especifican el tráfico están en la tabla 2.5, donde, como se ve, el tráfico satisface el modelo *Tenet* ( $X_{min}$ ,  $X_{ave}$ ,  $I$ ,  $S_{max}$ )

Parámetros de QoS	Descripción
$D_{m\acute{a}x}$	Límite superior de retraso del mensaje de emisor a receptor.
$Z_{m\acute{i}n}$	Límite inferior de retraso del mensaje de emisor a receptor.
$J_{m\acute{a}x}$	Límite superior de la variación del retraso ( <i>delay jitter</i> ).
$W_{m\acute{i}n}$	Límite inferior en la probabilidad de no pérdida, debido a una sobrecarga de los <i>buffers</i> .
Parámetros del tráfico	
$X_{m\acute{i}n}$	Tiempo mínimo entre mensajes.
$X_{ave}$	Tiempo medio entre mensajes.
I	Intervalo medio.
$S_{m\acute{a}x}$	Tamaño máximo del mensaje.

Tabla 2.5: Parámetros de calidad de servicio y tráfico.

### 2.9.4.- Disciplina de servicio

RTIP (*Real-Time Internet Protocol*) es el nivel de red de la *Tenet Suite*. Su principal función es la de transportar los paquetes para cumplir los requerimientos del canal correspondiente. En contraste con IP, este protocolo está orientado a la conexión. RTIP también realiza la planificación de los paquetes basándose en los parámetros de calidad de conexión de cada canal. Ya que todos los paquetes de una conexión RTIP siguen el mismo camino, los paquetes llegan siempre en el mismo orden.

La cabecera RTIP es de tamaño fijo para permitir un rápido proceso (figura 2.11). Para poder trabajar con IP, los cuatro primeros bits de la cabecera identifican el paquete como RTIP.

0	4	8	16	24
RTIP	Sin uso		Local ID	
Longitud del paquete			Número de secuencia	
Timestamp				
Reservado			Header checksum	

Figura 2.11: Formato del paquete en Tenet.

Campo	Descripción	Tamaño
Versión	Contiene identificador de paquete RTIP	4
Local ID	Identificador del canal	16
Longitud del paquete	Número de octetos del paquete	16
Número de secuencia	Número de secuencia del paquete	16
Timestamp	Tiempo en el que el paquete se recibió por el módulo RTIP del emisor.	32
Header checksum	Sólo hay <i>checksum</i> de la cabecera ya que no se asegura la integridad de los datos.	16

Tabla 2.6: Descripción de los campos de un paquete Tenet.

La entidad RTIP en cada nodo tiene el objetivo de asegurar que todos los paquetes son transportados para que cumplan sus requerimientos de calidad. Cuando se establece una conexión, RTIP es informado del máximo retraso  $d$  que puede tener un paquete en ese

nodo. RTIP asegura que un paquete llegado al nodo en un tiempo  $T$  es transmitido al siguiente nodo antes de  $T + d$ . RCAP informa además, de la cantidad de espacio de *buffer* asignado a la nueva conexión y del tiempo mínimo y medio entre mensajes ( $X_{min}$ ,  $X_{ave}$  y  $I$ ).

Cada entidad RTIP contiene dos módulos:

- **Módulo de control de tasa:** Este módulo controla el tráfico de cada conexión y lo ajusta de acuerdo con la especificación del tráfico. Así, si un paquete no se esperaba antes del tiempo  $T_e$ , y llega antes, entonces el paquete no se convierte en elegible hasta el tiempo  $T_e$ . Cuando es elegible se transfiere al módulo planificador.
- **Módulo planificador:** Se encarga de planificar los paquetes elegibles, asegurando que se cumplan los *deadlines*.

Este módulo también se encarga de llevar estadísticas. La implementación de este agente se realiza en el *kernel* del sistema y coexiste con TCP, UDP y IP. Se han implementado las siguientes disciplinas de servicio: *Delay-Earliest-Due Rate*, *Jitter-Earliest-Due-Rate* y *Rate-Controlled Static Priority*, aunque se puede implementar cualquier otra disciplina debido a su diseño modular

### 2.9.5.- Control de admisión

Ferrari introduce en 1989 un control de admisión para nuevos canales basado en el protocolo EDD [Ferrari89] que es el que se detalla en este punto utilizando el modelo *Tenet* de tráfico. El control de admisión de nuevos canales se realiza en los nodos y el receptor. Hay dos tests en cada nodo que comprueban si pueden dar la calidad de servicio pedida sin afectar al resto de los canales, y otro test en receptor que comprueba si el retraso total está por debajo del exigido.

En general, el funcionamiento de los tests es el siguiente: en los nodos se comprueba si se puede admitir el nuevo canal y se calcula el retraso mínimo que se puede dar al canal. Con la información de todos estos retrasos mínimos el receptor comprueba si se puede garantizar el retraso máximo exigido. Si es así, distribuye el retraso para cada nodo, aceptando el canal. Este esquema funciona si las disciplinas de servicio son *non-work-conserving*, en las que el retraso total de un paquete se puede obtener como la suma de los retrasos individuales en cada nodo.

Este apartado se va a centrar en los tests deterministas. Existen otros tests para un servicio estadístico que son más complejos y están fuera del ámbito de esta tesis.

### 2.9.5.1.- Test en los nodos

El objetivo de los tests en los nodos es asegurar que se puede dar servicio al canal sin afectar a la calidad de servicio del resto de canales. Cuando un nodo recibe una petición de establecimiento de canal se realizan dos tests en los nodos. El primero comprueba que se puede aceptar el canal y el segundo determina el retraso máximo del paquete permitido en el nodo.

El primero, denominado **Test determinista** verifica que hay suficiente poder de proceso para acomodar el nuevo canal sin perturbar al resto. Como la máxima utilización de un nodo por el canal  $i$ , cuyos paquetes tienen un tiempo de servicio máximo de  $t_i$ , es  $t_i/x_{min,i}$ , la condición es:

$$\sum_j \frac{t_j}{x_{min,j}} \leq 1 \quad (1)$$

donde la suma se extiende para todos los canales deterministas, incluido el que se establece.

El segundo, denominado **Test del retraso límite** determina si se puede evitar la saturación del planificador en el nodo, y si es así, el mínimo retraso a asignar al canal que se va a establecer para que el objetivo se cumpla.

Para determinar si la saturación del planificador en el nodo es posible se dividen los  $m$  canales del nodo en dos conjuntos:  $A$  es el conjunto de canales cuyo retraso límite en el nodo es menor que la suma de los tiempos de servicio de todos los canales, y  $B$  es el conjunto de aquellos canales cuyo retraso límite es el nodo es mayor o igual que esa suma:

$$A = \{i \mid i = 1, \dots, a; d_i < \sum_{j=1}^m t_j\}, \quad (2)$$

$$B = \{i \mid i = a + 1, \dots, m; d_i \geq \sum_{j=1}^m t_j\}, \quad (3)$$

Se numeran los  $a$  canales en  $A$  de acuerdo al orden en que se van a planificar. Asumiendo que:

$$x_{min,l} \geq \sum_{j=1}^m t_j \quad (l = 1, \dots, m) \quad (4)$$

que es lo mismo que decir que no llegarán más paquetes a un nodo que puedan interferir con la transmisión de aquellos que estamos considerando, entonces la saturación del planificador es imposible si y sólo si:

$$d_i \geq \sum_{j=1}^i t_j + \max_{i < k \leq m} t_k, \quad (i = 1, \dots, a) \quad (5)$$

Si la condición (4) no se cumple, se aplicará la siguiente fórmula al conjunto de canales que incluyen paquetes seguidos. Considerando que estos paquetes incrementan el  $\hat{a}t$ , se calcula el nuevo valor de esta suma  $(\hat{a}t)'$ , con la siguiente definición recursiva:

$$\begin{aligned} (\sum t)'_o &= \sum t, \\ (\sum t)'_{h+1} &= (\sum t)'_h + Y_k t_k, \quad (h = 0, 1, \dots), \quad Y_k = \begin{cases} 1 & \text{si } x_{\min, k} + d_k < (\sum t)' \\ 0 & \end{cases} \end{aligned} \quad (6)$$

Con la ecuación (5) se puede calcular el mínimo valor  $d_i$  con el que se cumple la condición de no saturable. Este valor se enviará al nodo receptor.

### 2.9.5.2.- Tests en el receptor

Cuando llega al receptor el mensaje de establecimiento de canal se realizan dos tests. El test del retraso comprueba si no se sobrepasa el retraso total:

$$D_i \geq \sum_n \min d_{i,n} + \sum_n l_{i,n} \quad (7)$$

Si se pasan todos los tests en los nodos, el receptor divide el retraso  $D$  entre los nodos atravesados por el canal, después de restar el retraso total en los enlaces de la ruta. Sea  $d_{i,n}$  el mínimo retraso asignado al canal  $i$  en el nodo  $n$ . Un paquete viajando en este canal y llegando al nodo en el tiempo  $t$  se le asignará un retraso máximo de  $t + d_{i,n}$ . Para satisfacer el retraso total, es suficiente satisfacer el límite  $d$  en cada nodo a través de la ruta. Los valores  $D_i$  se pueden redistribuir utilizando las siguiente fórmula:

$$d_{i,k} = \frac{\left( D_i - \sum_n \min d_{i,n} - \sum_n l_{i,n} \right)}{n} + \min d_{i,k} \quad (8)$$

### 2.9.5.3.- Tests adicionales

Estos tests son bastantes pesimistas. El grupo *Tenet* ha desarrollado otros tests más exactos en los nodos para distintos planificadores y especificación de tráfico. Existen tests exactos para FCFS, SP, EDF [Wregle96] y RCSP [Zhang94]. El gran problema que tienen estos tests es que necesitan un conocimiento exacto del tráfico, lo cual los hace complejos de implementar. Para solucionar este problema el grupo introdujo el concepto de envolvente empírica que se describirá en el siguiente capítulo.

## 2.10.- RSVP (*Resource ReSerVation Protocol*)

### 2.10.1.- Introducción

En respuesta a la creciente demanda de servicios integrados en Internet, la *Internet Engineering Task Force* (IETF) creó varios grupos de trabajo en 1993, entre ellos el de servicio integrados “*IntServ*” y el de definición del protocolo de reserva RSVP.

RSVP se ha diseñado para permitir a los emisores, receptores y *routers* de las sesiones de comunicación (tanto *multicast* como *unicast*) comunicarse con el resto para establecer una ruta que pueda soportar la calidad de servicio requerida [ZhangL93] [White97]. RSVP identifica una sesión por medio de una dirección de destino, un tipo de protocolo de transporte y un número de puerto de destino. RSVP no es un protocolo de encaminamiento; se usa meramente para reservar recursos a través de la ruta que se establezca por cualquiera de los protocolos de niveles inferiores.

El protocolo está definido por varias RFC (2205 a 2210), entre las que destaca la especificación funcional RFC2205 [Braden97]. Además existe un proyecto para una nueva versión del protocolo denominada RSVP2 de la *University of Southern California /Information Sciences Institute*[USCISI96]. También existen varias implementaciones de libre distribución para Linux, FreeBSD, etc. [Kuznetsov99]. Microsoft también da soporte recientemente a RSVP en Windows 2000 [Microsoft99] y CISCO ha anunciado *routers* con soporte RSVP.

Hay una extensión de RSVP para su uso en redes compartidas como LAN, que es denominada SBM (*Subnet Bandwidth Manager*). El problema en este tipo de redes compartidas es que el protocolo RSVP falla. Esto es debido a que sus mensajes deben pasar por varios dispositivos usando el nivel-2 (puentes, etc) que no conocen el protocolo RSVP de nivel 3 y superiores, con lo que no se pueden gestionar los recursos.

Aunque RSVP en principio es sólo un protocolo de reserva de recursos se suele asociar a las especificaciones de flujos definidas por el grupo *IntServ*, así como su control de admisión.

### **2.10.2.- Objetivos de diseño**

RSVP se ha diseñado con los siguientes objetivos:

1. Proporcionar la posibilidad de que **receptores heterogéneos** puedan hacer reservas de acuerdo a sus necesidades. No se debe asumir que todos los receptores tienen las mismas capacidades ni que requieran la misma calidad de servicio.
2. Debe adaptarse a las **variaciones de miembros** en grupos multidifusión. La conexión o desconexión de los miembros de un grupo debe ser dinámica.
3. Permitir a los usuarios **especificar sus necesidades** a nivel de aplicación para que los recursos reservados para un grupo multidifusión puedan reflejar con precisión los recursos necesitados por el grupo.
4. Permitir a los receptores **seleccionar entre varios canales**. El receptor debería poder seleccionar entre varias fuentes sin el riesgo de que la petición de este cambio fuera denegada, como podría ocurrir si se estableciese una nueva petición. Esto es posible si los recursos reservados por el receptor son reutilizados para la transmisión de esta fuente.
5. Debe **adaptarse a los cambios en los rutas** uni y multidifusión. RSVP no es un algoritmo de encaminamiento, utiliza el nivel de red para estos propósitos, pero si debe mantener un estado de las rutas.
6. **Controlar la sobrecarga** que produce el protocolo en la red para que no crezca linealmente o con el número de participantes.
7. Hacer el **diseño modular** para acomodar distintas tecnologías.

### **2.10.3.- Principios de diseño**

Para obtener los objetivos vistos en el punto anterior el diseño de RSVP se basa en seis principios básicos:

1. **Reserva iniciada por el receptor**. Los receptores escogen el nivel de servicio requerido y son responsables de iniciar y mantener la reserva activa mientras

quieran recibir datos. Esto es así, porque el receptor es quien conoce sus limitaciones y la calidad de servicio que recibe. Además, esto permite la gestión de peticiones heterogéneas.

2. **Filtro de paquetes.** La reserva de recursos en un *router* asigna ciertos recursos a la entidad que hace la reserva, pero no determina qué paquetes pueden usar estos recursos. Hay una función separada, llamado filtro de paquetes, que selecciona los paquetes que pueden usar estos recursos. Este filtro puede ser estático o dinámico y permite establecer varios modelos de reserva.
3. **Proporcionar varios estilos de reserva.** Por medio del filtro de paquetes se pueden definir diferentes modelos de reserva. Actualmente existen tres estilos: libre, filtro fijo y filtro dinámico.
4. **Mantener un estado (“soft-state”) de la red.** Durante una comunicación larga es posible que nuevos miembros se unan al grupo mientras que otros lo dejen, y las rutas pueden modificarse debido a cambios en la red. Por eso, RSVP debe mantener un estado de la red. Esta información se mantiene por medio de mensajes que periódicamente se envían para refrescar el estado. RSVP distingue dos clases de información en cada *router*: el estado de la ruta y el estado de la reserva.
5. **Control de sobrecarga del protocolo.** La sobrecarga de RSVP se determina por tres factores: el número de mensajes RSVP enviados, el tamaño de estos mensajes y las frecuencias de refresco de los mensajes de ruta y reserva. Para reducir la sobrecarga, RSVP funde los dos mensajes mientras atraviesan la red.
6. **Modular.** RSVP tiene interfaz con otros tres componentes en la arquitectura: (1) la especificación de flujo (`flowspec`) que se maneja a nivel de aplicación o sesión; (2) el protocolo de encaminamiento de red, que lleva los mensajes hasta los receptores; (3) el control de admisión en red, que realiza las decisiones basado en el `flowspec` que está en los mensajes de reserva.

En la figura 2.12. se muestra como encaja RSVP dentro de los *routers*, emisores y receptores.

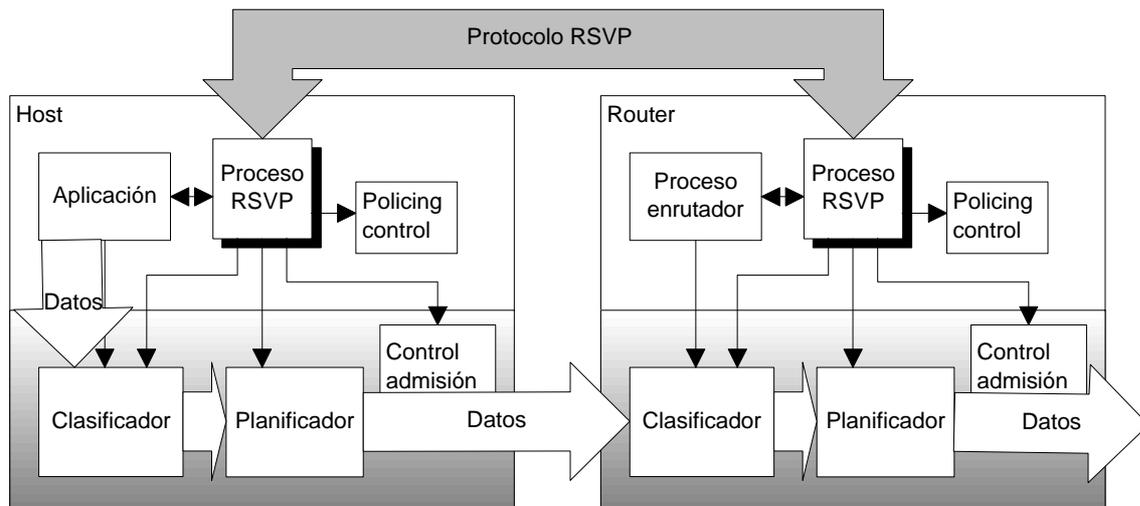


Figura 2.12: Módulo RSVP dentro del emisor y el router.

#### 2.10.4.- Clases de calidad de servicio

IETF ha considerado varias clases de calidad de servicio, aunque sólo dos han sido formalmente especificadas para RSVP: Servicio Garantizado (*Guaranteed Service*) [Schenker96] y Servicio de Carga Controlada (*Controlled-Load Service*) [Wroclawski96].

##### 2.10.4.1.- Servicio garantizado

Esta calidad de servicio está destinada para aplicaciones con requerimientos exigentes de tiempo real. Esta calidad asegura: un ancho de banda, un límite en el retraso y ninguna pérdida en las colas.

Como introducción, un flujo es descrito usando un modelo *token bucket* y dada esta descripción, cualquier elemento de la red (un *router*, una subred, etc.) calcula varios parámetros describiendo cómo va a manejar los datos del flujo. Combinando los parámetros de los distintos elementos que recorre el flujo, es posible calcular el retraso máximo que se producirá en el flujo.

Cada *router* caracteriza el servicio garantizado para un flujo determinado, asignando un ancho de banda  $R$ , y un espacio de memoria (*buffer space*)  $B$ , que representa los recursos que el flujo puede consumir. Esto representa que existe un ancho de banda  $R$  entre emisor y receptor. Así, para un servicio que siga el modelo de flujo perfecto con un cubo de capacidad  $b$  y tasa  $r$ , se puede calcular el límite del retraso como  $b/R$  siempre que  $R$  sea mayor  $r$ . Para permitir desviaciones sobre este modelo perfecto se añaden dos términos de error  $C$  y  $D$ ; con lo que el límite del retraso se convierte en  $b/R + C/R + D$ . Estos errores aparecen al trabajar con paquetes. Por ejemplo, cualquier paquete puede

experimentar un retraso debido a paquetes de su propia cola o debido a imprecisiones del planificador. El término  $C$  es el error dependiente de la tasa y representa el retraso que un paquete en el flujo puede experimentar debido a la tasa reservada. El término de error  $D$  es el error independiente de la tasa y representa el peor caso de variación de tiempo de tránsito a través del *router*.

Sin embargo, con el servicio garantizado se impone una cota superior a la tasa de transmisión que es la tasa pico del flujo  $p$ , que reduce los límites del retraso. Además, se tienen que tener en cuenta los efectos de la partición en paquetes en el flujo considerando el tamaño máximo del paquete,  $M$ . Esto nos permite disponer de un límite más preciso para el retraso:

$$Q_{delay} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M+C_{tot})}{R} + D_{tot} \quad (\text{caso } p > R \geq r) \quad (9)$$

$$Q_{delay} = \frac{(M+C_{tot})}{R} + D_{tot} \quad (\text{caso } R \geq p \geq r) \quad (10)$$

donde  $C_{tot}$  y  $D_{tot}$  representan el sumatorio de los términos de error  $C$  y  $D$  de cada *router* de la ruta. Es importante fijarse que la ecuación (9) no depende de los parámetros del flujo, debido a que como la reserva  $R$  es mayor que la tasa pico de la carga  $p$  no hay posibilidad de que los paquetes se encolen en la red.

Para que este retraso se cumpla se tiene que cumplir que el tráfico entrante esté conforme al algoritmo *token bucket*, es decir, el tráfico está limitado por la siguiente ecuación  $\min[pt, b+rt]$ .

Cada *router* necesita ser informado de las características del tráfico,  $T_{spec}$ , y del flujo con las características de las reservas realizadas,  $R_{spec}$ . Además, necesita los términos  $C_{sum}$  y  $D_{sum}$  que representan la suma de los términos de error  $C$  y  $D$  de cada *router* desde el origen del mensaje. Los parámetros  $T_{spec}$  y  $R_{spec}$  están enumerados en la tabla 2.7.

Para utilizar este esquema los *routers* tienen que aproximarse al modelo de flujo. Para ello, se pueden utilizar varios algoritmos de planificación como el WFQ (*Weighted Fair Queueing*) [Demers89], *Jitter-EDD* [Verma91] y *Virtual Clock* [ZhangL90].

<i>Parámetro Tspec</i>	<i>Descripción</i>	<i>Unidad</i>
p	Tasa pico del flujo.	Bytes/s
b	( <i>bucket depth</i> ) Tamaño del cubo.	Bytes
r	( <i>token bucket rate</i> )Tasa de transmisión del cubo.	Bytes/s
m	Tamaño mínimo de un paquete.	Bytes
M	Tamaño máximo de paquete.	Bytes
<i>Parámetros Rspec</i>		
R	Ancho de banda.	Bytes/s
S	<i>Slack term</i> . Diferencia entre el retraso deseado y el obtenido usando una reserva de ancho de banda R.	μs

Tabla 2.7: Parámetros Tspec y RSpec.

Concretamente, con el planificador WFQ, los parámetros  $C_i$  y  $D_i$  pueden ser calculados de la siguiente forma:  $D_i$  es igual al *MTU* (*Maximum Transmission Unit*) del enlace dividido por el ancho de banda del enlace, con la condición de que  $M$  debe ser menor que el mínimo *MTU* del path. El valor de  $C_i$  se asume que es  $M$  para considerar la fragmentación en paquetes. Esto es:

$$D_{tot} = \sum D_i = \sum \frac{MTU_i}{AB_i} \quad C_{tot} = \sum C_i = \sum M \quad (11)$$

Respecto al tamaño del *buffer* a reservar en los nodos, el tamaño necesario en cada nodo es  $b + C_{sum} + D_{sum}r$ , donde  $C_{sum}$  y  $D_{sum}$  son la suma de todos los parámetros  $C_i$  y  $D_i$  anteriores al nodo.

#### 2.10.4.2.- Servicio de carga controlada

Esta clase de servicio no proporciona garantía firme de que se cumpla el servicio requerido. Se puede indicar un Tspec para la calidad servicio requerida, aunque no es necesario incluir el parámetro de tasa pico  $p$ . Si el flujo es aceptado, el *router* intenta ofrecer un servicio equivalente a un flujo “*best-effort*” en una red ligeramente cargada. La diferencia es que este flujo no se deteriora aunque aumente la carga de la red.

Esta clase de servicio está destinada a aplicaciones que puedan tolerar una cierta pérdida de información o retraso, siempre que se mantenga en límites razonables.

#### 2.10.5.- Funcionamiento de RSVP

##### 2.10.5.1.- Mensajes de establecimiento de ruta

Los mensajes primarios usados por RSVP son el mensaje Path que tiene su origen en el emisor, y el mensaje Resv, que tiene su origen en el receptor:

- Mensaje Path: Su objetivo es primero instalar un estado del encaminamiento inverso a través de la ruta, y segundo proporcionar a los receptores información sobre las características del tráfico a enviar y de la ruta para que se puedan hacer las peticiones de reserva adecuadas.
- Mensaje Resv: Realizan las peticiones de reserva a los *routers* a lo largo del árbol de distribución entre receptores y emisores.

En resumen (véase la figura 2.13), el emisor envía al receptor un mensaje Path. Si éste llega correctamente al destino y la reserva es posible, se envía un mensaje Resv de vuelta al emisor. Este mensaje reservará de forma definitiva los recursos en la red por los nodos en los que vaya pasando. Cuando este mensaje llegue al emisor, la transmisión puede comenzar.

Los mensajes se pueden transportar dentro de datagramas IP (en otros sistemas se tendría que utilizar UDP). Cada mensaje Path incluye la información descrita en la tabla 2.8.

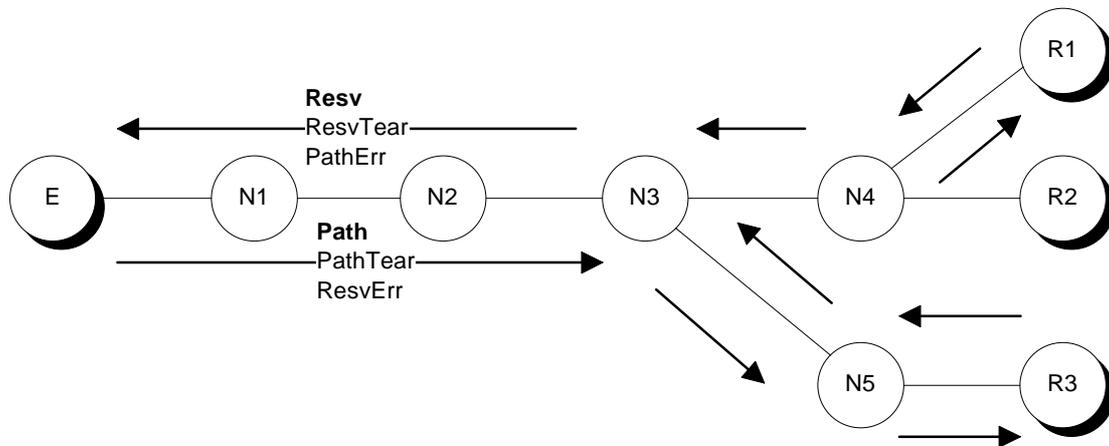


Figura 2.13: Dirección de los mensajes RSVP.

Path message	Descripción
Pop	Dirección del último nodo RSVP capaz de enviar este mensaje Path
Sender Template	Contiene la dirección IP del emisor y opcionalmente el puerto.
Sender Tspec	Define las características del tráfico
Adspec	Opcional. Contiene información que es actualizada por cada <i>router</i> de la ruta. Contiene la información OPWA ( <i>One Pass With Advertising</i> )

Tabla 2.8: Parámetros del mensaje path.

Cada receptor debe primero unirse a un grupo multidifusión para empezar a recibir los mensajes Path. Esta gestión de los grupos multidifusión está fuera del ámbito del protocolo RSVP.

### 2.10.5.2.- Proceso y propagación de los mensajes Path

Cada *router* del árbol de distribución intercepta los mensajes Path y chequea su validez. Si se detecta un error, el *router* enviará un mensaje PathErr para informar al emisor que no puede realizar ninguna acción apropiada. Asumiendo que el mensaje es válido, el *router* hace lo siguiente:

- Actualiza el estado de la entrada de la ruta para el emisor identificado en el Sender Template. Si no existe la ruta la crea. El estado de la ruta incluye: Sender Tspec, dirección, Phop del anterior *router* y opcionalmente el Adspec. La dirección Phop es necesaria para encaminar los mensajes Resv en el sentido contrario.
- Actualiza los contadores de limpieza de rutas a su valor inicial.

RSVP incorpora un protocolo de mensajes con refresco periódico para mantener un estado de los *routers* intermedios para proporcionar fiabilidad y seguridad. Para ello, cada entrada en el *router* tiene un contador asociado que cuando llega a cero elimina la conexión. Para que esto no ocurra, las rutas activas tienen que recibir un refresco por medio del mensaje Path a intervalos regulares. Este periodo debe ser bastante menor que el tiempo de los contadores de limpieza para que no produzcan desconexiones innecesarias.

Aparte de la eliminación de las rutas de forma automática, RSVP incluye el mensaje PathTear para eliminar la ruta de forma activa.

### 2.10.5.3.- Objeto ADSPEC

El objeto Adspec se puede incluir en los mensajes Path para enviar a los receptores las características de la ruta de comunicación establecida. Esta información puede ser usada por los receptores para determinar el nivel de reserva requerido con el objetivo de obtener la calidad de servicio deseada. Este objeto consiste en una cabecera de mensaje, un fragmento con los parámetros generales por defecto (*Default General Parameters*), y al menos uno de los dos fragmentos del Servicio Garantizado o Servicio de carga controlado:

Parámetros generales por defecto	Descripción
Latencia mínima de la ruta	Suma individual de las latencias de los enlaces.
Ancho de banda de la ruta	El mínimo de los anchos de banda de los enlaces.
<i>Global bit break</i>	Este bit lo pone a uno el emisor. Se pone a cero en el caso de que el paquete pase por un <i>router</i> que no soporte RSVP para indicar que la información contenida puede no ser válida.
Contador de salto IS	Se incrementa en uno por cada <i>router</i> RSVP.
PathMTU	<i>Path maximum transmission unit.</i> (es el mínimo de las MTU's de los enlaces individuales de la ruta).
<b>Parámetros de servicio garantizado</b>	
$C_{tot}$	Valor total del parámetro C de emisor a receptor.
$D_{tot}$	Valor total del parámetro D de emisor a receptor.
$C_{Sum}$	Suma compuesta del valor C hasta el anterior <i>router</i> .
$D_{Sum}$	Suma compuesta del valor D hasta el anterior <i>router</i> .
<i>Break bit</i>	Mismo funcionamiento que <i>Global bit break</i> .
Parámetros generales	Estos parámetros son opcionales pero si se incluyen sustituyen los valores por defecto. Puede ser utilizado por los <i>routers</i> que tengan unos requerimientos específicos.
<b>Parámetros de servicio controlado</b>	
<i>Break bit</i>	Equivalente al anterior.
Parámetros generales	Equivalente al anterior.

Tabla 2.9: Objeto *Adspec*.

Este paquete no puede ser nunca fragmentado, por lo tanto, el valor de  $M$  de una petición de reserva no puede ser mayor que  $PathMTU$ . Toda esta información será actualizada por cada *router* RSVP a lo largo de la ruta.

### 2.10.6.- Control de admisión

El control de admisión de un nuevo canal se realiza en el receptor. Cuando el receptor recibe un mensaje *Path* extrae los siguientes parámetros del *Sender Tspec*:  $r$ ,  $b$ ,  $p$  y  $m$ . Además, también extrae del objeto *Adspec* los siguientes parámetros: latencia mínima de la ruta,  $C_{tot}$ ,  $D_{tot}$ ,  $PathMTU$  y ancho de banda de la ruta.

El límite requerido para el retraso de cola,  $Q_{delreq}$ , se calcula restando la latencia mínima de la ruta del valor del retraso de emisor a receptor requerido por la aplicación receptora. El receptor realizará un chequeo inicial evaluando la ecuación (10) para  $R$  igual a la tasa pico  $p$ . Si el resultado es mayor o igual que  $Q_{delreq}$  se utilizará esta fórmula para calcular el valor mínimo de  $R$  necesario para satisfacer  $Q_{delreq}$ ; sino se utilizará la ecuación (9) para este propósito. Este valor mínimo de  $R$  se obtiene insertando  $Q_{delreq}$  en la ecuación (9) o (10) con los valores determinados de  $C_{tot}$ ,  $D_{tot}$ ,  $r$ ,  $b$ ,  $p$  y  $M$ . Si el valor  $R$  excede el ancho de banda obtenido del *Adspec* recibido se reducirá. El receptor entonces puede crear una especificación de la reserva, *Rspec*, que contiene el valor  $R$  de ancho de banda

que se reservará en cada *router* y un término *slack* que será inicialmente cero. Rspec forma parte del mensaje Resv cuyos parámetros están en la tabla 2.10:

Resv message	Descripción
Estilo de reserva	Indica el estilo de reserva a utilizar. Puede ser FF, SE o WF.
FilterSpec	Especificación de filtro para identificar a los emisores.
Flowspec	Se compone del Rspec y la especificación de tráfico, Tspec.
ResvConf	Opcionalmente se envía un objeto de confirmación conteniendo la dirección IP del receptor.

Tabla 2.10: Mensaje Resv.

Este mensaje se envía de vuelta por la ruta que ha recorrido. Por cada *router* que pasa de vuelta, los mensajes se pueden fusionar con otros mensajes Resv con la misma interfaz, de acuerdo a una serie de reglas que dependen del estilo de reserva, obteniendo un nuevo Flowspec y FilterSpec. Cada *router* realiza además las siguientes acciones :

- El Flowspec se pasa al módulo de control del tráfico que aplica el control de admisión para determinar si la reserva se acepta.
- Si la reserva es denegada, se envía un mensaje ResvErr.
- Si la reserva es aceptada, el estado de las reservas se actualiza de acuerdo con los parámetros FilterSpec y FlowSpec. La reserva puede ser mezclada con otras reservas de acuerdo con el estilo de reserva, y con esto se creará un nuevo mensaje Resv.

### 2.10.7.- Modelos de reserva de recursos

Como se vio en los objetivos de diseño, RSVP modela una reserva por medio de dos componentes, una asignación de recursos y un filtro de paquetes. La asignación de recursos especifica qué cantidad de recursos son reservados, mientras el filtro de paquetes selecciona qué paquetes pueden usar los recursos. Esta distinción y la posibilidad de cambiar el filtro de paquetes dinámicamente permite a RSVP ofrecer varios estilos de reserva. Un estilo de reserva captura los requerimientos de comunicaciones del nivel de aplicación. Por ahora se han definido tres modelos de reserva:

- **Libre (Wildcard):** Este modo indica que cualquier paquete con destino a un grupo multidifusión asociado puede utilizar los recursos reservados. Esto permite hacer una única asignación de recursos a través de todas las rutas de distribución del grupo.

- **Filtro Fijo** (*Fixed Filter*): Este modo indica que mientras dure la conexión el receptor sólo recibirá paquetes de las fuentes indicadas en la petición de reserva original.
- **Filtro dinámico** (*Dynamic Filter*): Se permite durante la conexión modificar la función de filtro. Esto permite la posibilidad de dinámicamente seleccionar un canal entre las distintas fuentes. Esto requiere que se asignen los recursos suficientes para manejar el peor caso que es cuando todos los receptores pidan de diferentes fuentes.

### 2.10.8.- Tipos de encaminamiento para RSVP

Aunque se ha visto que RSVP no es un protocolo de encaminamiento si que hay cuatro problemas que se deben tratar con el protocolo de encaminamiento :

1. Encontrar una ruta que soporte la reserva de recursos, es decir, que tenga implementado el protocolo RSVP en todos los *routers*.
2. Encontrar una ruta que tenga la suficiente capacidad disponible para un nuevo flujo. Se puede optar por dos formas diferentes de encontrar esta ruta. Una podría ser la de modificar los protocolos de encaminamiento y gestionarlos de acuerdo a un mecanismo de control del tráfico. Alternativamente, el protocolo de encaminamiento podría ser rediseñado para proporcionar múltiples rutas alternativas, y en la reserva podría intentarlo en cada una de las rutas.
3. Adaptarse a un fallo de ruta. Cuando un nodo falla, el encaminamiento adaptativo encontrará una ruta alternativa. El refresco periódico de RSVP automáticamente hará una reserva en la nueva ruta. Pero, la nueva reserva puede fallar porque no haya suficiente capacidad disponible en la nueva ruta. Esto es un problema de dimensionamiento y calidad de la red, que no puede ser solucionado por los protocolos de encaminamiento o reserva.
4. Adaptarse a un cambio de ruta (sin fallo) . Los cambios de ruta pueden ocurrir sin que se produzcan fallos. Aunque RSVP podría usar las misma técnicas de reparación que las descritas en el punto 3, esta solución podría producir una merma en la calidad de servicio. Podría ocurrir que si el control de admisión falla en la nueva ruta, el usuario verá una degradación del servicio innecesaria y caprichosa, ya que la ruta original está todavía funcional. Para evitar este problema, se sugiere un mecanismo de fijado de rutas (*route pinning*) en el que las rutas se mantienen fijas mientras sean viables.

RSVP está actualmente diseñado para trabajar con cualquier protocolo de encaminamiento disponible sin modificación. Esto puede provocar que se produzcan ciertas degradaciones en la calidad de servicio al no cumplirse los anteriores requerimientos. Se espera que las futuras generaciones de protocolos de encaminamiento incluirán mecanismos que, en conjunción con RSVP, resolverán los problemas enumerados.

### ***2.11.- Arquitectura de servicios diferenciados.***

En este punto se describe brevemente el estado actual de la arquitectura de servicios diferenciados y su relación con la de servicios integrados. Como se ha comentado en el capítulo anterior, IETF creó recientemente el grupo de Servicios Diferenciados (*DiffServ group*) con el objeto de normalizar y crear estándares para la implantación de una arquitectura de servicios diferenciados en Internet. Esta arquitectura ha sido concebida con el objetivo de proporcionar calidad de servicio de una forma escalable [Blake98]. Actualmente existen varios proyectos de implementación de redes basadas en esta arquitectura como AQUILA [Koch00] y TEQUILA [Goderis00].

La diferenciación de servicio se basa en la definición de una serie de clases de servicio que la red tratará de forma distinta y que permiten obtener distintos niveles de calidad. En general, una arquitectura de servicios diferenciados se basa en tres principios básicos:

- **Clasificación:** Se marcan los paquetes en los nodos de entrada a la red de acuerdo a la clase de servicio requerida.
- **Diferenciación:** Este marcado de los paquetes permite determinar como son tratados dentro de la red, y por tanto obtienen un servicio diferenciado.
- **Contención:** La obtención de la calidad de servicio se basa en controlar la congestión o carga de la red. Esto se realiza conformando los paquetes en los nodos de entrada de acuerdo a los requerimientos o reglas de cada clase de servicio.

En la figura 2.14 se muestra un ejemplo de transmisión entre un emisor y receptor usando una red de servicios diferenciados. En primer lugar, el emisor establece una clase de servicio con la que realizar la transmisión. Esta clase de servicio implicará un marcado en un nodo cercano al emisor (clasificador: *first hop node*). Estos paquetes marcados entrarán a la red por un nodo de entrada (*ingress node*) que realizará la contención del flujo. Esto significa que si los paquetes marcados no cumplen con los patrones

determinados por la clase de servicio serán almacenados o eliminados. Dentro de la red, los nodos internos realizarán un tratamiento diferenciado en función de la clase de servicio que viene determinada en la marca de cada paquete. La salida de la red se realizará por el nodo de salida (*egress node*) que a su vez puede enlazar con otra red.

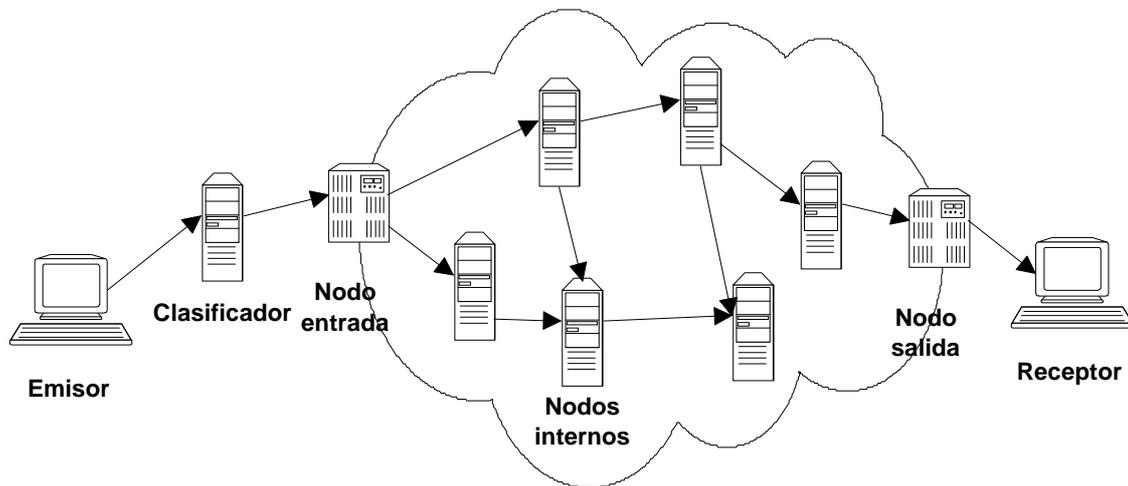


Figura 2.14: Red de servicios diferenciados.

Para poder distinguir las clases de servicio, IETF ha propuesto el uso de un *byte* especial dentro de la cabecera del protocolo IP que se denomina el campo de servicios diferenciados (*DS Field*). Este byte contiene 6 bits denominados *DiffServ Code Point* (DSCP) [Nichols98]. El comportamiento de los nodos ante una clase de servicio se denomina "comportamiento por salto" (*per-hop behavior: PHB*). Actualmente IETF ha definido dos clases de servicio (dos PHB), basándose en el trabajo de Dave Clark y Van Jacobson [Nichols99], aparte de la clase *best-effort* establecida actualmente en Internet:

- **Servicio premium** (o *Expedited forwarding*) [Jacobson99]: Su objetivo es obtener pequeños retrasos siempre que el tráfico origen cumpla con un patrón determinado. Es como una línea dedicada en la que no se puede exceder el ancho de banda requerido.
- **Servicio seguro** (*assured service*) [Heinanen99]: Este servicio asigna al cliente un ancho de banda determinado pero sin garantías. Ofrece un servicio mejor que el *best-effort* de Internet.

Cada PHB está definiendo cómo se comporta el nodo ante una determinada clase de servicio y no cómo se implementa. Por ejemplo, en el servicio premium el comportamiento de un nodo ante el tráfico agregado de esta clase de servicio es que la tasa de salida debe ser igual o mayor a la tasa de entrada. Además debe ser independiente de la carga de la red.

Este comportamiento puede ser implementado por medio de una cola con prioridades o un esquema CBQ [Floyd95].

Se define también, cómo el tráfico es limitado a la entrada de la red. En este sentido se utiliza un conformador *token bucket* y el tráfico que no esté conforme es eliminado. En el servicio seguro, este comportamiento es diferente y el tráfico no conforme no es marcado y se trata como *best-effort*.

Para la gestión y control del tráfico de la red se añade un elemento denominado gestor de ancho de banda (*Bandwith Broker: BB*) [Nichols99]. Este elemento tiene dos responsabilidades:

- Controlar la carga de la red e informar a los nodos de entrada y salida (*ingress/egress nodes*) de la situación de la red para poder contener el tráfico a la entrada.
- Negociar con las redes adyacentes el control de los flujos.

La relación entre los servicios integrados y diferenciados es un tema de estudio actualmente. IETF está trabajando en este campo y en general se propone tratar la red de servicios diferenciados como un nodo dentro de la red de servicios integrados [Bernet00]. Se está hablando por tanto, de servicios integrados sobre servicios diferenciados.

Esto significa que toda la red se comportará como un nodo de la red de servicio integrados como se muestra en la figura 2.15. Para ello, los nodos de entrada y salida de la red gestionarán los mensajes RSVP. Esto es, cuando reciban el mensaje *Path* asociarán al canal una clase de servicio y generarán los términos de error  $C$  y  $D$  para toda la subred como si fuera un nodo (valores  $C_{ds}$  y  $D_{ds}$  de la figura). Si el canal es admitido, cuando el nodo de entrada reciba el mensaje *Resv* asociará los paquetes del canal a la clase de servicio requerida para su marcado.

También está en estudio el mapeo de las clases de servicio *IntServ* de Servicio Controlado y Garantizado a las clases *DiffServ* [Wroclawski00]. Esto definirá la reserva a realizar y cómo calcular los términos de error  $C_{ds}$  y  $D_{ds}$ .

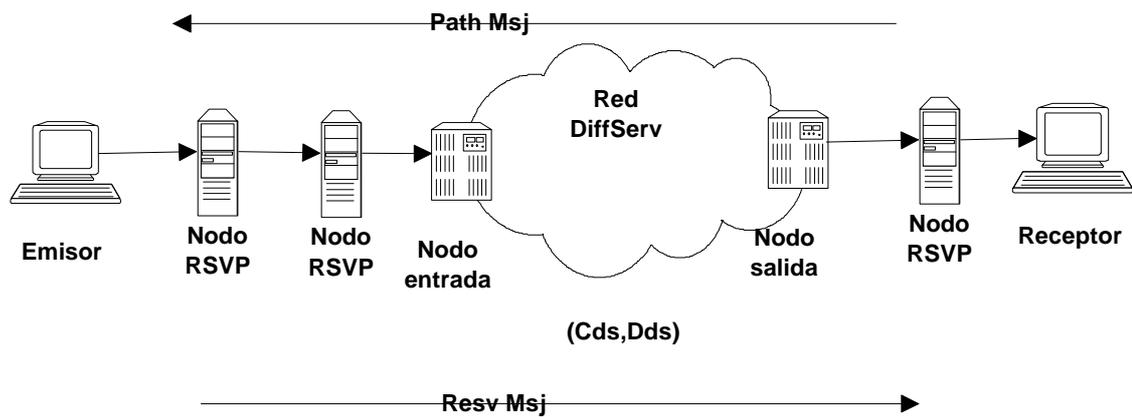


Figura 2.15: Servicios integrados sobre una red de servicios diferenciados.

## 2.12.- Conclusiones

En la primera parte de este capítulo se han descrito los componentes más importantes para la transmisión en tiempo real. Estos son:

- La especificación del tráfico.
- Planificadores y disciplinas de servicio.
- El control de admisión.
- Mecanismo de reserva de recursos.

En general, estos componentes son dependientes entre sí. Por ejemplo, el control de admisión depende de la especificación del tráfico y la disciplina de servicio del nodo.

En el resto del capítulo se han descrito tres tipos de redes que presentan enfoques distintos para tratar la transmisión en tiempo real. ATM es una red comercial que ofrece una solución completa para la transmisión en tiempo real a nivel de red. La suite *Tenet* es una red desarrollada para probar distintos aspectos de la transmisión en tiempo real. El protocolo RSVP junto a la especificación de tráfico garantizado de IETF resuelven parte de los problemas de la transmisión en tiempo real como son la reserva de recursos y el control de admisión.

En la tabla 2.11 se describe cómo implementan los distintos elementos las redes estudiadas. Cuando no se indica nada es que la red no especifica qué modelo utilizar.

	Modelo de tráfico	Disciplina de servicio	Control de admisión	Reserva de recursos
<b>ATM</b>	Varios : CBR, VBR, rt-VBR, etc.		CAC	
<b>Tenet Suite</b>	Modelo Tenet ( $X_{min}$ , $X_{ave}$ , $I$ , $S_{max}$ )	RCSP, Delay-EDR, Jitter-EDR	Depende de la disciplina. En nodo y receptor.	Una pasada.
<b>RSVP/IETF</b>	<i>Token bucket</i>	( <i>WFQ o similares</i> )	En receptor.	Protocolo RSVP. Doble pasada.

Tabla 2.11: Cuadro de las componentes principales de las redes estudiadas.

Al final del capítulo se ha descrito brevemente el modelo de servicios diferenciados. Se ha visto, que hay una fácil integración de esta red dentro de un modelo de servicios integrados. Se comprueba que el modelo de servicios integrados es adecuado para integrar los distintos elementos que atraviesa un canal desde un emisor a un receptor, ya sean redes particulares (ATM, LAN, etc ) o redes de servicios diferenciados.

## 3.

# Descripción del tráfico

### 3.1.- Características del tráfico

#### 3.1.1.- Introducción

Normalmente, el tráfico transmitido por la red se comprime antes de ser enviado a la red. La compresión vendrá determinada por la forma en que se pueda introducir el tráfico en la red. En este sentido se puede distinguir:

- *CBR (Constant Bit Rate)*: Tasa de transmisión constante. La red reserva una tasa de transmisión constante a la aplicación, con lo que el objetivo de la compresión va a ser obtener una tasa de transmisión constante o con pocas variaciones para que sea eficiente.
- *VBR (Variable Bit Rate)*: La red acepta una especificación del tráfico con una tasa de transmisión variable.

Se pueden distinguir los siguientes modos de operación [*Lakshman98*]:

- *Unconstrained (U)-VBR*: Codificador en bucle abierto, no hay interacción con la red (Por ejemplo codificación MPEGx, H.26x).
- *Shaped (S)-VBR*: La salida del codificador es conformada antes de introducirla en la red, como por ejemplo con un modelo *leaky bucket*, pero sin afectar al codificador.
- *Constrained (C)-VBR*: El codificador se ajusta a los parámetros de salida a la red, como por ejemplo a los parámetros del *leaky bucket*. A diferencia del anterior, se intenta optimizar la codificación para ajustarse al máximo a los parámetros definidos por la red.

- *Feedback (F)-VBR*: El codificador se ajusta a los parámetros actuales de la red, de tal forma que se puede variar la calidad de transmisión. Un ejemplo de este esquema es el reproductor de Internet Real-Player.

### 3.1.2.- Codificación y compresión

El tráfico multimedia que se transmite en la red va codificado y normalmente comprimido con el objeto de reducir la cantidad de información a transmitir. Como la codificación del tráfico es determinante en la generación del tráfico en la red, se realiza una breve descripción de los mecanismos de codificación usualmente usados.

Se pueden clasificar los esquemas de codificación en la siguiente jerarquía:

- *Compresión individual de tramas (Intraframe compresión)*: En este modo, cada trama es comprimida y codificada independientemente, como por ejemplo *Moving JPEG [Wallace91]*.
- *Compresión individual y conjunta de tramas (Intraframe and Interframe compresión)*: Cada trama es comprimida individualmente y conjuntamente como en los estándares MPEG<sup>2</sup> y ITU H.261. Este esquema permite enfoques de codificación más complejos permitiendo una cierta interacción con el nivel de transporte. Por ejemplo, en el tráfico MPEG se podrían enviar las tramas I, P y B por varios canales con distintas prioridades.
- *Compresión por capas (Layered compresión)*: La idea es codificar en varios niveles. Este esquema permite optimizar la información en función del destino. Se puede codificar a un primer nivel en blanco y negro, luego la información adicional del color y luego aumentar a alta definición. El cliente escogería qué niveles se transmiten en función de su capacidad de representación.

Como se ha visto en el punto anterior, la codificación puede ser independiente de la red, con lo que se obtiene un tráfico codificado, como puedan ser los ficheros MPEG o bien puede ser codificado *on-line*, teniendo en cuenta las características de la red.

---

<sup>2</sup> Ver apéndice 1 sobre una descripción del tráfico MPEG.

### 3.1.3.- Tráfico en ráfagas y dependencias a largo plazo

Un tema importante en la descripción del tráfico es el concepto del tráfico en ráfagas (*burstiness*), exhibido, por ejemplo, en tráfico como: vídeo comprimido, transferencia de ficheros, etc. Básicamente, las ráfagas ocurren cuando las llegadas parecen que forman visualmente grupos, esto es, existen momentos con tiempo entre llegadas muy reducidos y otros momentos con tiempos mucho más largos.

El modo más simple de medir este efecto, es calcular la proporción de la tasa pico de transmisión con respecto a la tasa media. Una medida más elaborada es el coeficiente de variación, definido como la proporción de la desviación estándar a la media  $c_A = \sigma[A_n] / E[A_n]$  de los intervalos entre llegadas. Finalmente, el parámetro Hurst [Hurst51][Leland94] puede ser usado como una medida de la variabilidad del tráfico basado en el concepto de tráfico autosemejante (*auto-similar*).

Las dependencias a largo plazo, que son reflejadas en coeficientes de autocorrelación con un decaimiento lento o subexponencial, son esenciales para predecir las prestaciones de las redes. Esto es importante, porque autocorrelaciones positivas pueden explicar el fenómeno del tráfico en ráfagas. En consecuencia, son necesarios modelos que capturen esta naturaleza del tráfico.

## 3.2.- Modelado del tráfico

### 3.2.1.- Concepto de modelo de tráfico

Los modelos de tráfico<sup>3</sup> son empleados fundamentalmente de dos maneras: como parte de un modelo analítico, o para realizar una simulación por eventos. En este apartado se hace una breve descripción de los modelos de tráfico y sus características, no siendo el objetivo describirlos formalmente ni profundizar en sus bases estadísticas. En general, existe dificultad en encontrar la caracterización apropiada del tráfico que capture las dinámicas de la fuente.

Un tráfico simple consiste en la llegada de entidades discretas (paquetes, celdas), con intervalos definidos (tiempo entre llegadas  $A_n$ ). Un tráfico compuesto consiste en la llegada de más de una entidad en un instante  $T$ . Además, es útil incorporar la noción de

---

<sup>3</sup> Aunque en el capítulo anterior se introdujo el concepto de modelo de tráfico o flujo como la especificación de cómo se introduce el tráfico en la red, en este apartado, el modelo de tráfico se refiere a los modelos analíticos o estadísticos que intentan caracterizar el tráfico.

carga en la descripción del tráfico. La carga es un concepto general que describe la cantidad de trabajo que acarrea al sistema la llegada de una entidad.

### 3.2.2.- Modelos de tráfico

Los modelos han sido utilizados para describir distintos tipos de tráfico, como vídeo (MPEG), audio y transferencias de ficheros. A continuación se describen brevemente los modelos de tráfico más utilizados [Beran95] [Rose95b] [Casilari99] [Frost94] [Heyman96]:

- **Renewal Traffic Models:** Los valores  $A_n$  son independientes, distribuidos idénticamente, pero su distribución se permite que sea general. Los modelos de colas han asumido históricamente este tipo de tráfico. Este modelo tiene un grave inconveniente, la función de autocorrelación de  $\{A_n\}$  es cero para todos los intervalos de tiempo distintos de cero. Entre estos modelos están los *Procesos Poisson* y *Procesos Bernoulli*.
- **Cadenas de Markov:** Estos modelos introducen dependencia entre los valores  $A_n$ . Por lo tanto, pueden potencialmente capturar la variabilidad del tráfico, por su autocorrelación distinta a cero en  $\{A_n\}$ .
- **Modelos de tráfico fluido:** El paradigma de tráfico fluido ve el tráfico como un fluido, caracterizado por una tasa de flujo (por ejemplo bits por segundo). Estos modelos asumen que las fuentes son en ráfagas (del tipo “ON-OFF”). Mientras en el estado OFF no existe tráfico, en el estado ON el tráfico llega de forma determinista a una tasa  $\lambda$ .
- **Modelos de tráfico autorregresivo:** Estos modelos son usados normalmente para ajustar la función empírica de autocorrelación, pero generalmente no pueden ajustar la distribución marginal como el DAR (*Discrete Autoregressive*). Una mejora de este modelo es el TES (*Transform Expand-Sample*) que proporciona otro método para capturar tanto la autocorrelación y la distribución marginal de forma simultánea.
- **Modelos de tráfico autosemejante:** Estudios recientes de tráfico de alta calidad y alta resolución han revelado un nuevo fenómeno con importantes implicaciones en el modelo, diseño y control de redes. En estudios sobre redes *Ethernet* [Leland94] y sobre vídeo VBR [Beran95][Rose95a] parece que el tráfico es estadísticamente autosemejante. El fenómeno autosemejante (o fractal) exhibe similitudes estructurales en todas las escalas de tiempo. El

parámetro clave para caracterizar este fenómeno es el llamado parámetro Hurst,  $H$ , que indica el grado de autosemejanza de un tráfico dado. Entre estos modelos están los ruidos gaussianos fraccionarios o FGN (*Fractional Gaussian Noise*), los procesos ARIMA fraccionarios (FARIMA) y el modelado *Wavelet* [Casilari99]

Además, se pueden considerar dos clases de modelos [Lucantoni94]: modelos para tráficos multiplexados y modelos para tráficos simples. En los primeros se capturan los efectos de la multiplexación estadística de las fuentes y predicen que la superposición de varias fuentes de tráfico muy variable se suaviza. Estos modelos son útiles en el diseño, gestión y optimización de la red. En cambio, los modelos para tráficos simples son usados para estudiar qué tipos de parámetros de tráfico pueden ser usados para negociar con la red, predecir la calidad de servicio, admisión en la red, etc.

Para concluir, todos los modelos tienen sus pros y contras, y se tiene que tener cuidado en qué tipo de modelo escoger, en función del tipo de simulación o análisis a realizar.

### 3.3.- Suavizado

#### 3.3.1.- Técnicas de suavizado

El tráfico a ráfagas dificulta el diseño de protocolos de comunicaciones para tiempo real que logren un alto grado de utilización de recursos [Salehi98]. Como principio general de diseño, cuanto menos variable es una carga más fácil es de manejar. Hay tres técnicas básicas para el suavizado de la carga:

1. Suavizado por multiplexación temporal: Se puede obtener introduciendo un *buffer* por canal a lo largo del *path*. Esto puede suavizar la tasa pico del canal, pero al coste de introducir retraso.
2. Suavizado por agregación (multiplexación estadística): Ocurre cuando distintos canales comparten un mismo recurso (ancho de banda, *buffer*) donde cada uno tiene unos requerimientos de calidad de servicio estadístico. El teorema del límite central dice que los requerimientos de ancho de banda de los canales agregados convergen a una distribución normal cuando el número de canales es grande. Este suavizado, que es estadístico por naturaleza, no introduce retraso.
3. Suavizado por preprocesado (*work-ahead*): Se puede obtener enviando datos antes de tiempo, de tal modo que se reduzca la variabilidad de la tasa de

transmisión. Este esquema tiene dos restricciones: a) la información tiene que estar disponible y b) el cliente tiene que tener suficiente *buffer* para recibirlo. Este método no introduce retraso.

El rendimiento de la red depende fundamentalmente de la tasa pico y las características del tráfico admitido, y por tanto, el tráfico suavizado óptimamente requiere menos recursos.

### 3.3.2.- Suavizado en la red

Al introducir un tráfico en la red se produce ya un suavizado al almacenarse en los *buffers* de la red, así como al seguir la disciplina de servicio (*token bucket* o *leaky bucket*). En el modelo *leaky bucket* este suavizado se produce antes de introducir los paquetes en la red, ya que la tasa máxima de salida es  $r$ . En cambio con *token bucket* el suavizado se realiza en la red en el caso de que la red esté congestionada y los paquetes tengan que almacenarse en la red<sup>4</sup>.

En la figura 3.1 se puede ver un modelo de red con *token bucket*. El tráfico de entrada está definido por la función  $E(t)$ . Como se vio en el punto 2.3 el tráfico es introducido por la red y dependiendo de la carga de los nodos llegará de una forma u otra al receptor  $R(t)$ . La carga de los nodos impondrá cual es el suavizado. Si no hay carga en la red, esto implicará que el tráfico no se almacena en los nodos con lo que el tráfico del receptor es equivalente al del emisor. Si hay mucha carga en los nodos, entonces los paquetes se almacenarán en la red, lo que implica un retraso.

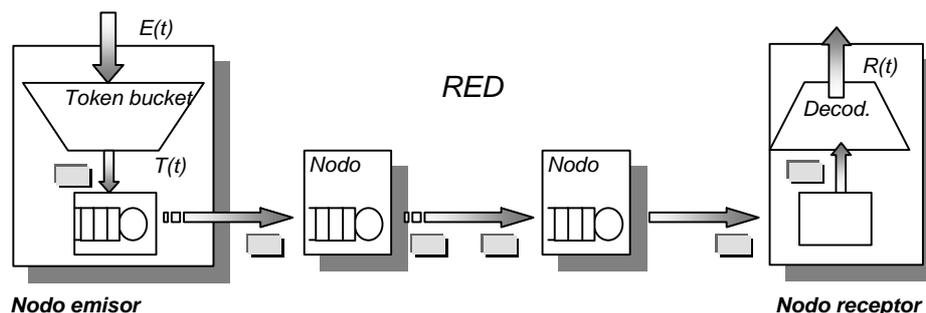


Figura 3.1: Elementos del suavizado en la red

<sup>4</sup> Esto en la práctica se suele cumplir aunque analíticamente no se puede demostrar [Zhang98].

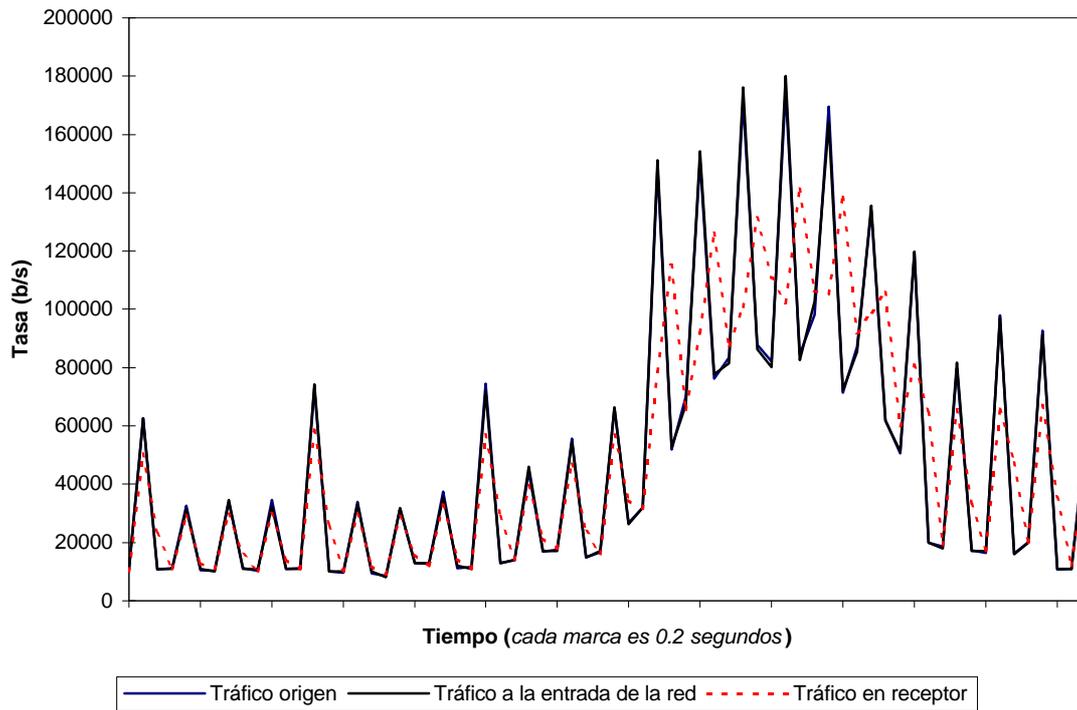


Figura 3.2: Ejemplo de suavizado en la red. Variación del tráfico en la red.

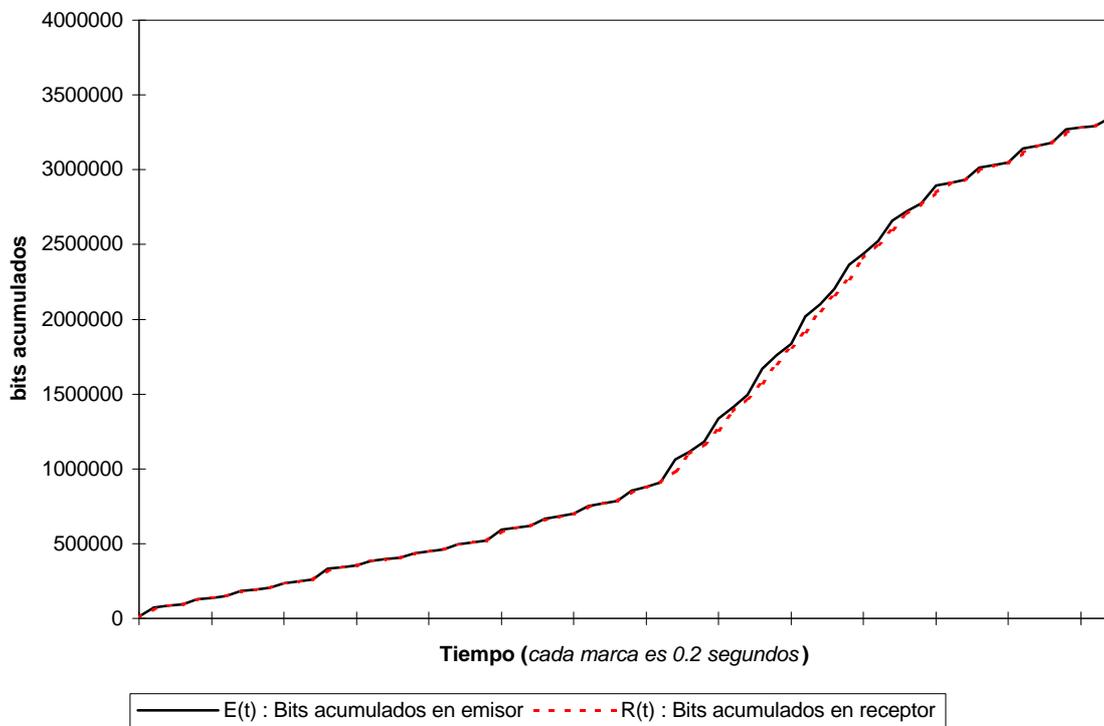


Figura 3.3: Ejemplo de suavizado en la red. Funciones  $E(t)$  y  $R(t)$ .

En la figura 3.2 se puede ver un ejemplo de suavizado en la red, simulando una red de 10 nodos bastante cargada. Como se ve, el tráfico tiende a suavizarse disminuyendo los picos. En la figura 3.3 se ven las funciones  $E(t)$  y  $R(t)$  para el mismo ejemplo. Estas gráficas son resultado de simulaciones realizadas con RTNetSim para probar los esquemas del capítulo 6 con una red de 10 nodos. Los parámetros de la red están descritos en ese capítulo y el apéndice B.

### 3.4.- *Envolvente empírica*

En este apartado se describen dos conceptos introducidos por el grupo *Tenet* utilizados ampliamente en esta tesis, como son las funciones de tráfico limitado y la envolvente empírica [Knightly94][Knightly95]. Estas funciones permiten caracterizar el tráfico de forma determinista mediante cotas superiores y permiten realizar un análisis del retraso máximo de un tráfico.

En general, para poder determinar el máximo retraso de un paquete, se necesita una descripción determinista del tráfico. Se podría utilizar la tasa pico de transmisión pero esto provocaría una utilización ineficiente de la red. Cuanto más se acerque el modelo de la carga a la real, más información dispondrá la red para planificar de forma adecuada los paquetes. De esta forma, aunque la suma de las tasas pico de todas las conexiones sea mayor que la velocidad del enlace, todavía se podrá proporcionar límites al retraso.

El modelo que se utilice debe tener las siguientes propiedades:

- Debe ser una caracterización del peor caso de la fuente para proporcionar un límite superior absoluto de la llegada de los paquetes de la fuente.
- Debe ser parametrizable para que la fuente pueda especificar eficientemente el tráfico.
- Debe caracterizar el tráfico lo más aproximadamente posible.

Se define la función  $A(t)$  como el número de bits transmitidos para un tiempo  $t$ . Esta función nos da una visión de los bits transmitidos, pero es dependiente del tiempo. Sin embargo, para algunos test de admisión sería necesario calcular todas las posibilidades en función del valor de  $t$ . Como en la práctica es inviable calcular todos los casos para cada realización del tráfico de entrada, es mejor encontrar el peor caso del tráfico de entrada, y comprobar que se puede planificar.

Se puede determinar el peor caso, por medio de una función límite del tráfico  $A_j^*(t)$  que denota el máximo número de bits que puede transmitir el canal  $j$  en un intervalo de

tiempo  $t$ . Si el tráfico de un canal está determinado por la función  $A$  tal que  $A[t, t+t]$  representa la transmisión en un intervalo de tiempo  $[t, t+t]$ , la función  $A^*(t)$  será un límite superior de  $A$  para todos los valores de  $t \geq 0$  y todos los intervalos de tiempo  $t$  si se cumple:

$$A[t, t+t] \leq A^*(t) \tag{1}$$

Cualquier función  $A^*(t)$  que satisface la propiedad se denomina “función de tráfico limitado”. Esta función proporciona un límite independiente del tiempo de  $A$ , por lo que el flujo está acotado para cada intervalo de longitud  $t$ .

Además, se define la función “envolvente empírica”  $E^*(t)$ , que es la función de tráfico limitado más exacta para una función  $A$ :

$$E^*(t) = \max_{t>0} A[t, t+t] \quad \forall t > 0 \tag{2}$$

donde se cumple siempre que:

$$A^*(t) \geq E^*(t) \quad \forall t > 0 \tag{3}$$

Por ejemplo, si el gráfico de la figura 3.4 representa la tasa de transmisión de un flujo, la función  $A(t)$ , la envolvente empírica  $E^*(t)$  y una función  $A^*(t)$  están representadas en la figura 3.5.

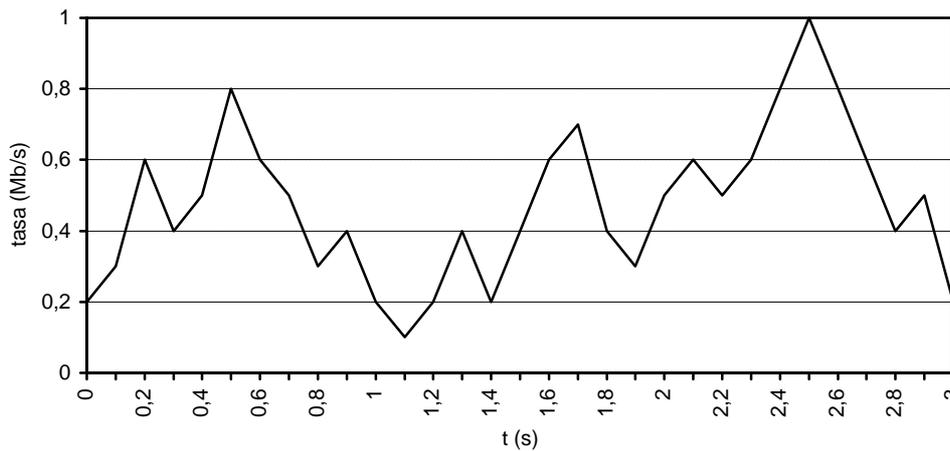


Figura 3.4: Tráfico de tasa variable.

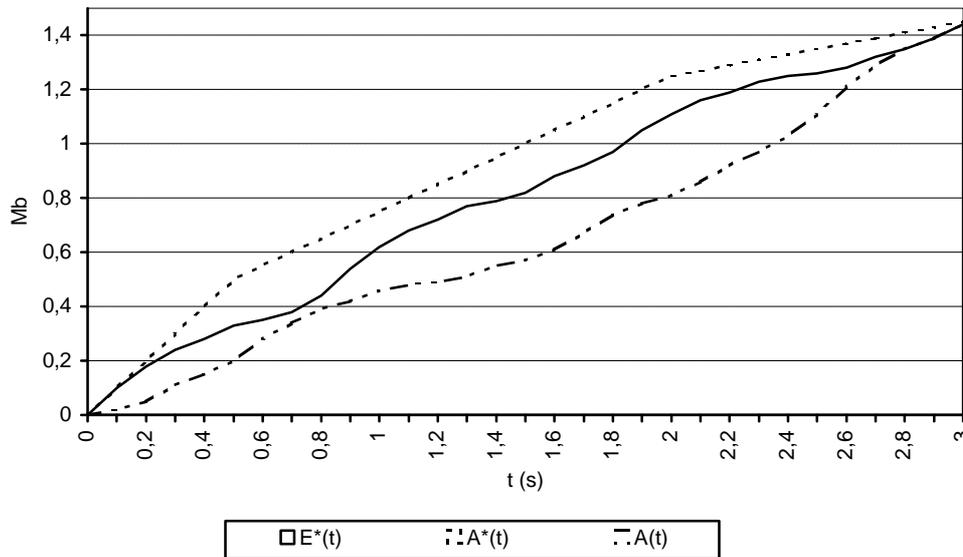


Figura 3.5: Función de tráfico limitado y envolvente empírica.

En la práctica, una fuente puede especificar el flujo por medio de un modelo parametrizado. Este modelo define una función que limita el tráfico. En el ejemplo de la figura 3.5 la función de tráfico limitado  $A^*(t)$  está definida por la siguiente función por intervalos:

$$A^*(t) = \begin{cases} t & 0 < t \leq 0.5 \\ 0.5 + (t - 0.5) * 0.5 & 0.5 < t \leq 2 \\ 1.25 + (t - 2) * 0.2 & 2 < t \leq 3 \end{cases}$$

Es interesante comentar que L. Cruz utiliza también un tráfico limitado que se denomina LBAP (*Linear Bounded Arrival Process*) [Cruz91a] [Cruz91b]. Esta limitación es la base para la obtención de cotas máximas en el retraso para redes en tiempo real.

### 3.5.- Comparación de tráfico

#### 3.5.1.- Introducción

En este punto, se introduce un mecanismo que permite comparar distintas especificaciones de tráfico y obtener sus parámetros. Este método está basado en las funciones de tráfico limitado, y como se verá, es bastante intuitivo.

En general, un problema que se plantea en la transmisión en tiempo real es cómo obtener los parámetros del modelo del tráfico. Si se dispone del tráfico (por ejemplo el

fichero MPEG de una película), antes de realizar su transmisión es necesario calcular los parámetros del modelo de tráfico que utilice la red (siguiendo el ejemplo, obtener los parámetros *token bucket*  $(b,r,p)$  a partir del vídeo MPEG a transmitir). Esto es fundamental, debido a que si los parámetros calculados son infraestimados (valores de  $r$  y  $b$  bajos), se corre el riesgo que el tráfico en un determinado momento sobrepasase el modelo y provoque congestión o pérdida de información en la red, o bien, si los parámetros son sobrestimados (altos valores de  $r$  y  $b$ ), se provocará un uso excesivo de los recursos en la red.

Si se desconoce exactamente el patrón del tráfico (por ejemplo en una transmisión en directo), estos parámetros habrá que estimarlos en función de la información que se pueda obtener del tipo de tráfico a transmitir (siguiendo el ejemplo, un debate televisivo y un partido de fútbol tendrían parámetros distintos). El método que se describe en este punto sólo es válido si se dispone del tráfico, como suele ocurrir en la transmisión de vídeo bajo demanda.

La obtención de los parámetros se realiza en dos fases:

1. Se calculan las funciones de tráfico limitado para cada modelo descrito en el capítulo anterior (*Tenet*, *RSVP-IETF*, *ATM*). Cada modelo de tráfico está definiendo una función de tráfico limitado. Además, el uso de estas funciones permite comparar los distintos modelos de tráfico existentes.
2. Usando la envolvente empírica sobre el tráfico a transmitir se obtienen los parámetros ajustando gráficamente las funciones de tráfico limitado.

Para probar este método se utilizan dos cargas ficticias que representan tráficos CBR y VBR. Con estos tráficos se calcula la envolvente empírica y se ajustan a las funciones de tráfico limitado. A partir de ajustar las funciones se obtienen directamente los parámetros de cada modelo de tráfico.

### 3.5.2.- Funciones de tráfico limitado

Como se ha comentado, cada modelo en realidad está definiendo una función de tráfico limitado  $l(t)$  que indica el número máximo de bytes que se puede enviar en cada intervalo de tiempo  $t$ . Los modelos que se han visto en el capítulo anterior son el servicio garantizado RSVP-IETF que sigue un modelo *token bucket*, el modelo *Tenet* y la especificación de tráfico VBR de ATM. Además, se añade un modelo de flujo constante para poder comparar. Los parámetros de cada modelo están definidos en la tabla 3.1.

Parámetros	Descripción	Unidad
<b>RSVP-IETF:</b>		
p	Tasa pico del flujo.	Bytes/s
b	Tamaño del cubo ( <i>bucket depth</i> ).	Bytes
r	Tasa de transmisión del cubo ( <i>token bucket rate</i> ).	Bytes/s
m	Tamaño mínimo del paquete.	Bytes
M	Tamaño máximo de paquete.	Bytes
<b>Modelo Tenet</b>		
$X_{\min}$	Tiempo mínimo entre mensajes.	s
$X_{\text{ave}}$	Tiempo medio entre mensajes.	s
I	Intervalo medio para $X_{\text{ave}}$ .	s
$S_{\max}$	Tamaño máximo del mensaje.	Bytes
<b>ATM – Tráfico VBR</b>		
PCR	Tasa pico de celdas.	Celdas/s
SCR	Tasa sostenido de celdas.	Celdas/s
MBS	Factor burstiness de transmisión.	Celdas
<b>Flujo constante (CBR)</b>		
$X_{\min}$	Tiempo mínimo entre mensajes.	s
$S_{\max}$	Tamaño máximo del mensaje.	Bytes

Tabla 3.1: Modelos usados y sus parámetros.

En la tabla 3.2 se muestran las funciones de tráfico limitado para los modelos estudiados. En el modelo *Tenet* y CBR se tienen en cuenta los efectos de la cuantificación en paquetes, mientras que en ATM y RSVP se omite. La figura 3.6 muestra las funciones de tráfico limitado para RSVP-IETF, *Tenet Suite* y CBR.

Modelo	Función de tráfico limitado
RSVP-IETF	$l(t) = b + rt$
Tenet Suite	$l(t) = S_{\max} + \min\left(\frac{S_{\max}}{x_{\min}} t, \frac{S_{\max}}{x_{\text{ave}}} I\right) \quad t \leq I$
ATM – VBR	$l(t) = \min(PCR \cdot t, MBS) \quad T < MBS / SCR$
CBR	$l(t) = S_{\max} + \frac{S_{\max}}{x_{\min}} t$

Tabla 3.2: Funciones de tráfico limitado.

Si se representa el número máximo de celdas en función del tiempo se tienen las funciones de tráfico limitado para CBR y VBR en ATM en la figura 3.7. Por ejemplo, para un tráfico VBR definido con los siguientes parámetros PCR = 50000 celdas/s, SCR = 10000 celdas/s y MBS = 2000 celdas se tendrá una función como la representada en la figura 3.8.

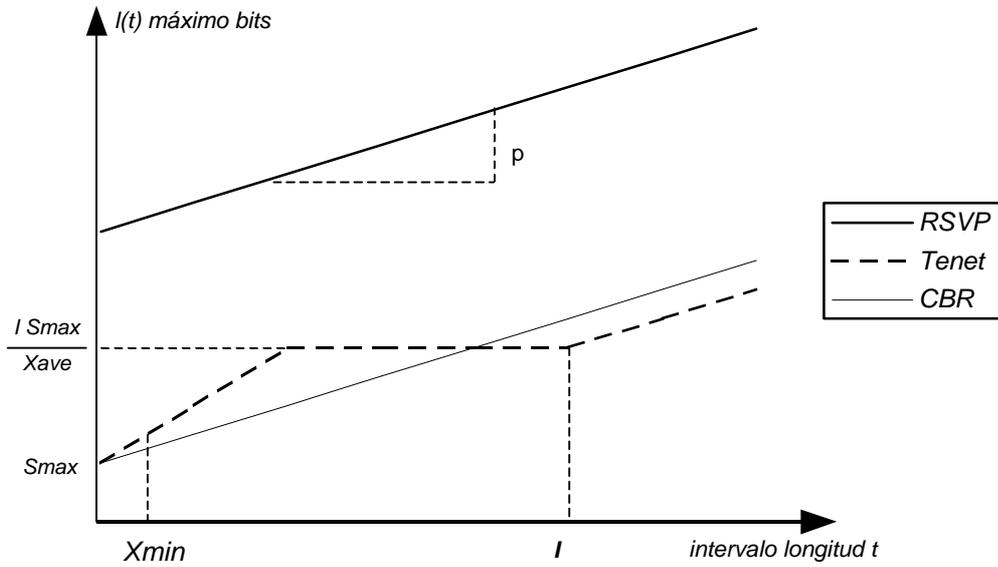


Figura 3.6: Funciones de tráfico limitado para RSVP, Tenet y CBR.

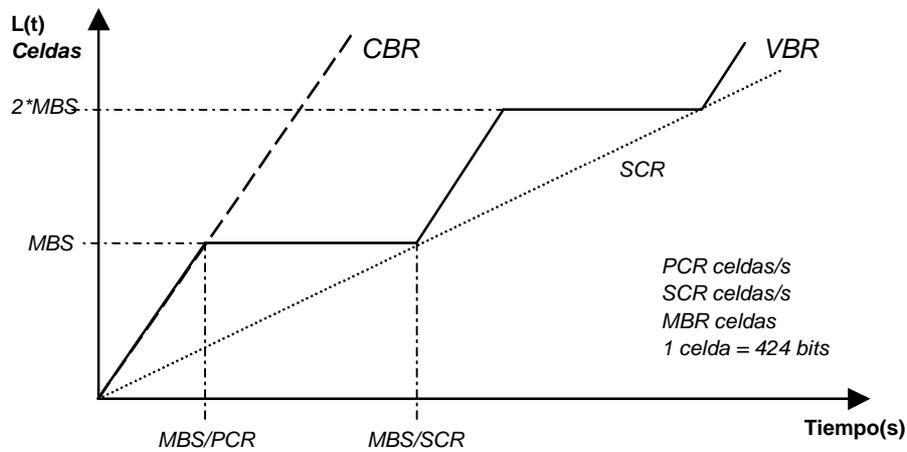


Figura 3.7: Funciones de tráfico limitado en ATM.

Como se puede ver, el tráfico VBR de ATM es equivalente al modelo *Tenet* (las curvas VBR de la figura 3.7 y de *Tenet* de la figura 3.6 son equivalentes). Se puede establecer, por tanto, la siguiente relación entre los parámetros de ambos modelos:

$$MBS \times C = I \frac{S_{max}}{X_{ave}}, \quad \frac{MBS}{SCR} = I, \quad PCR \times C = \frac{1}{X_{min}} S_{max} \tag{4}$$

donde \$C\$ es el tamaño de la celda (52 bytes).

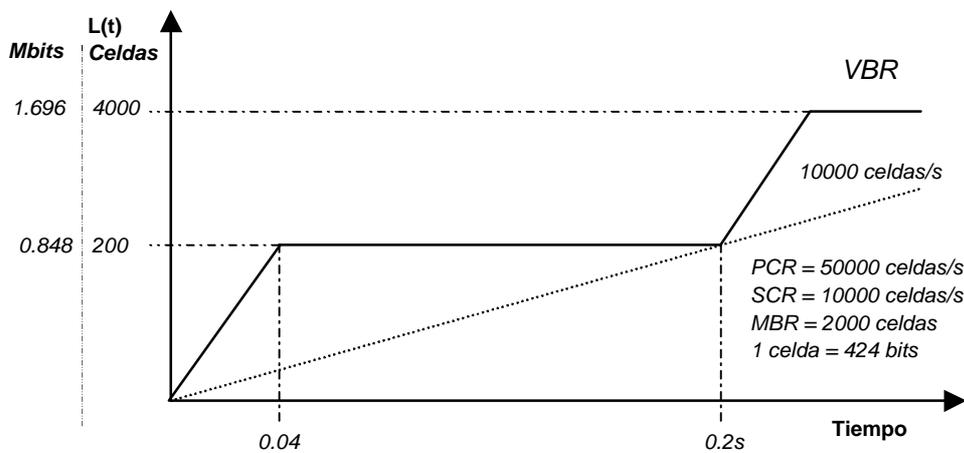


Figura 3.8: Ejemplo de función de tráfico limitado para ATM.

### 3.5.3.- Obtención de los parámetros

En este punto, se obtienen los parámetros para cada modelo a partir de los tráfico. El método es el siguiente: dado un determinado tráfico se obtiene su envolvente empírica. Ya que la envolvente empírica es la función de tráfico limitado más ajustada al tráfico, cualquier función del tráfico que la contenga estará limitando el tráfico sin pérdida. Por lo tanto, se intenta ajustar por encima las funciones de tráfico limitado de tal forma que la diferencia sea mínima. Este sistema permite obtener varios parámetros en función de cómo se ajusten las funciones. Como se verá en capítulos posteriores, del ajuste de estas funciones dependerá la reserva de recursos en la red.

Para esta comparación se van a distinguir dos tipos de tráfico:

- **CBR:** “*Delay Sensitive Constant bit-rate*”: Tráfico en tiempo real con una tasa de transmisión constante. Por ejemplo, audio con muestreo fijo sin comprimir.
- **VBR:** “*Delay Sensitive Variable bit-rate*”: Tráfico en tiempo real con una tasa de transmisión variable. El ejemplo más claro de este tipo de tráfico es la transmisión de vídeo comprimido en formato MPEG, en el que cada imagen puede tener una tasa de compresión bastante variable.

Para todos los flujos se va utilizar un valor de  $I = 3s$  y  $S_{max} = M = 10000$  bits.



Los valores resultantes para cada descripción se obtienen directamente a partir de la figura 3.10 y están en la tabla 3.3. Los parámetros ATM se obtiene a partir de los de *Tenet* usando las equivalencias (4) descritas en el apartado anterior.

Modelo	Parámetros
Tenet Suite	$X_{ave} = 0.0286s$ $X_{min} = 0.01s$
CBR	$X_{min} = 0.01s$
RSVP-IETF	$b = 0.3 Mb$ $r = 0.36 Mb/s$ $p = 1 Mb/s$
ATM-VBR	PCR = 2404 celdas/s, MBS = 2522 celdas, SCR = 841 celdas/s

Tabla 3.3: Parámetros para tráfico VBR.

### 3.5.3.2.- Flujo 3: CBR

Este flujo tiene una tasa de transmisión constante o con mínimas variaciones tal como muestra la figura 3.11.

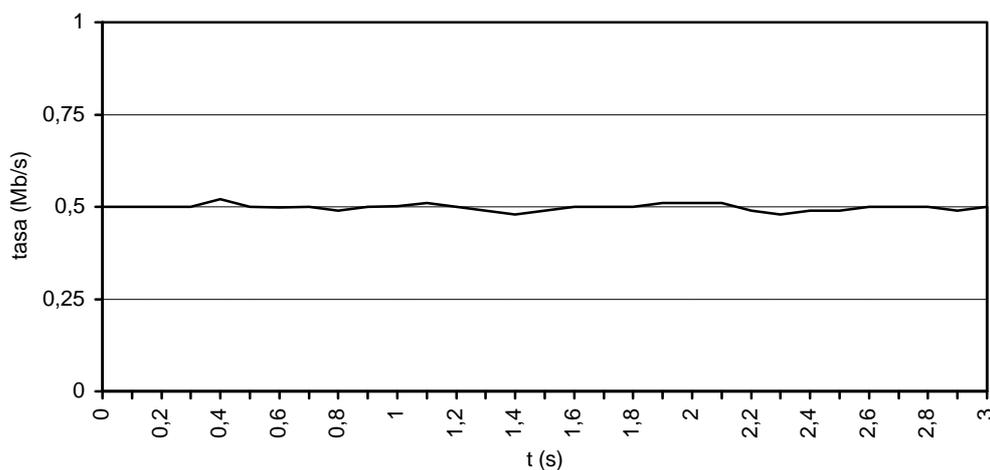


Figura 3.11: Tráfico CBR.

Para este caso, las funciones de tráfico limitado son equivalentes como se muestra en la figura 3.12. Los parámetros obtenidos son equivalentes (tabla 3.4) y no aportan ninguna ventaja con respecto al modelo CBR.

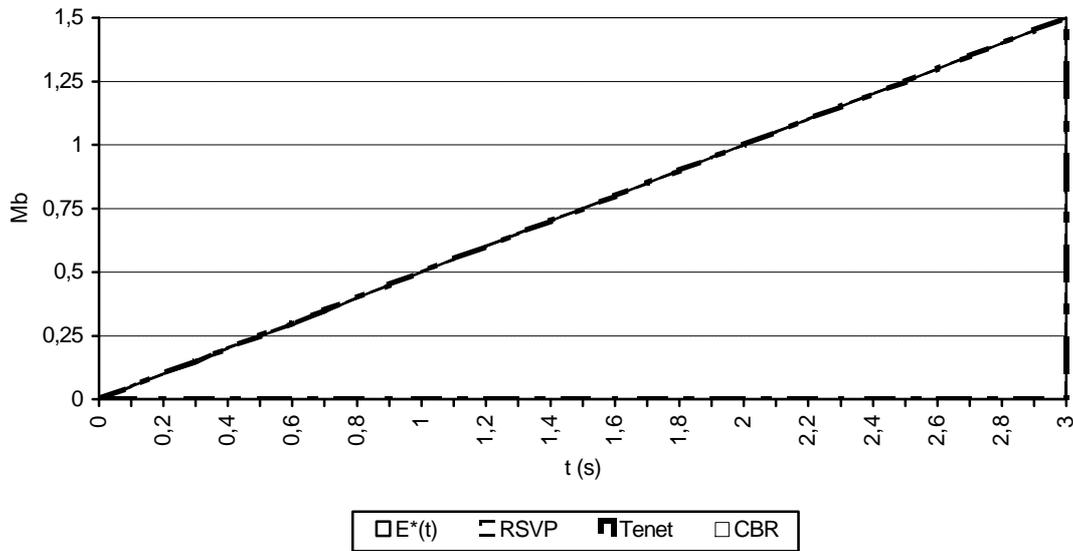


Figura 3.12: Funciones de tráfico limitado para CBR.

Modelo	Parámetros
Tenet Suite	$X_{ave} = X_{min} = 0.02s$
CBR	$X_{min} = 0.02s$
RSVP-IETF	$b = 0 Mb$ $r = 0.5 Mb/s$ $p = 0.5 Mb/s$
ATM-VBR	PCR = 1202 celdas/s, MBS = 3606 celdas, SCR = 1202 celdas/s

Tabla 3.4: Parámetros para tráfico CBR.

### 3.6.- Conclusiones

En este capítulo se han descrito las características fundamentales del tráfico y su modelado. Se ha descrito el concepto de las funciones de tráfico limitado y la envolvente empírica que son la base de desarrollos posteriores en la tesis.

Utilizando estas funciones se ha introducido un método para comparar los modelos de tráfico de cualquier especificación de tráfico. Esto permite también, que a partir de un tráfico determinado se pueden obtener los parámetros del modelo del tráfico. Esto es muy importante, ya que permite ajustar los parámetros a la carga a enviar de tal forma que no estén sobreestimados y el tráfico siga siempre el modelo sin producir congestión o pérdida de información.



## 4.

# Reserva óptima de ancho de banda

### 4.1.- Introducción

En este capítulo y el siguiente se aborda la optimización de la reserva de recursos en la red, concretamente el ancho de banda reservado en los enlaces y la memoria en los nodos. En ambos casos, se usa el concepto de los puntos envolventes que se introduce en este capítulo. Para realizar estas optimizaciones se impone una condición: se tiene que conocer el tráfico que se va a transmitir. Por ello, se plantea su utilización en la transmisión bajo demanda, siendo también aplicable a otros tipos de transmisión.

La motivación para el diseño de servidores de ficheros específicos para la transmisión de medios como vídeo y audio está bien definida [Rangan91][Anderson92]. En este sentido, la transmisión de vídeo o audio bajo demanda (*media-on-demand*) será un hecho habitual en los próximos años. Para ello existirán una serie de servidores que permitirán acceder en tiempo real a cualquier película, vídeo, música, etc., que nos interese [Deloddere94][Crowcroft99]. El cliente recibirá la película por medio de un ordenador o por medio de un decodificador (STB: *Set Top Box*). El grado de servicio puede variar entre un modelo de reproducción interactivo (como un reproductor de vídeo: con pausa, repetición, cámara lenta, etc.) a un modelo no interactivo en el que la película se transmite a una determinada hora sin posibilidad de control.

Como se ve en la figura 4.1, se tiene un servidor con un conjunto de discos donde están almacenados los medios. Cuando un cliente quiere ver una película se conectará a este servidor y seleccionará la que desee. En este momento, se iniciará una transmisión de vídeo que discurrirá por el canal establecido, desde el servidor al cliente. Para establecer el canal habrá que reservar una serie de recursos en la red para asegurar que la transmisión se realiza con una determinada calidad de servicio. Es importante, para ahorrar recursos en la red, que la reserva de los recursos sea óptima. Dado que la película estará almacenada siempre en el servidor, se conoce de antemano las características del tráfico a transmitir.

Un problema que se plantea al realizar una transmisión de vídeo<sup>5</sup>, es cómo calcular los parámetros que describen el tráfico. En general, se tiene un tráfico que sufre variaciones de ancho de banda en función del tiempo, lo que se denomina tráfico VBR (*Variable Bit Rate*). A la hora de transmitir este tráfico, se necesita un modelo que limite esta transmisión. El modelo de tráfico más usado es el *leaky bucket* ( $s, r$ ) donde el emisor no puede transmitir en el periodo  $[0, t]$  más de  $s + rt$  bits. A partir de esta restricción y con los parámetros de la red se han desarrollado un conjunto de ecuaciones que determinan el retraso de emisor a receptor a partir de una reserva realizada en los nodos. Entre estas ecuaciones, las más conocidas son las basadas en el trabajo de Parekh y Gallager [Parekh92][Parekh93].

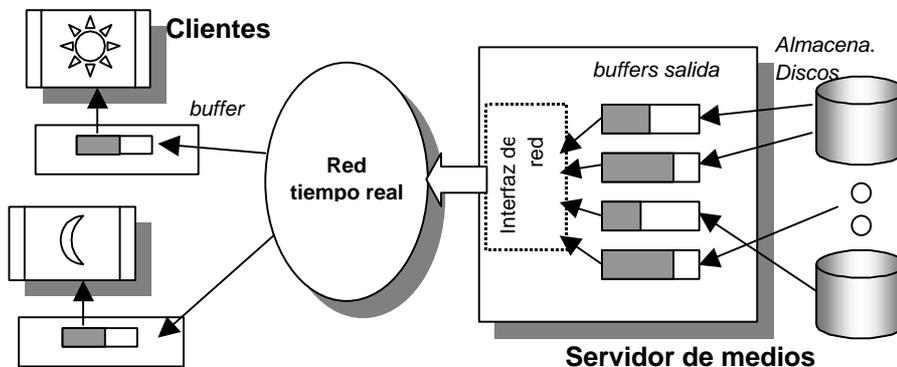


Figura 4.1: Vídeo bajo demanda.

Como se ha descrito en el capítulo 2, con el objetivo de la transmisión en tiempo real en Internet, IETF ha trabajado en la especificación de una serie de protocolos para servicios integrados como RSVP [Braden97], con una clase de servicio garantizado [Schenker96] que utiliza una descripción del tráfico basada en *token bucket*.

Trabajos anteriores han estudiado cómo optimizar la calidad del vídeo para una especificación del tráfico *leaky bucket*. El objetivo de los trabajos [Hsu97][Chen97] es obtener los algoritmos de codificación óptimos seleccionando conjuntamente las tasas de codificación y transmisión en la red. Otras técnicas se basan en el suavizado del tráfico VBR con el objetivo de reducir la tasa pico. El suavizado se puede conseguir enviando información antes de tiempo. Algunos trabajos presentan algoritmos de suavizado óptimos dados los parámetros de red como el tamaño de cubo o el ancho de banda reservado [Salehi98][Reibman95][Rexford99]. Pero tal como se indica en [Lakshman98] en vez de

<sup>5</sup> Aunque en esta tesis se trabaja particularmente con vídeo, los resultados son aplicables a cualquier tráfico VBR.

obtener una optimización conjunta de las tasas de codificación y de red se puede realizar en dos pasos. Esto es, se puede obtener el vídeo codificado optimizando la calidad de la imagen, para posteriormente, en la transmisión seleccionar la tasa de transmisión óptima. Para esta segunda fase, se presenta un nuevo método muy rápido para obtener la tasa óptima de transmisión para un vídeo ya codificado. A continuación se detallan, con un poco más de profundidad las soluciones propuestas.

#### **4.1.1.- Antecedentes y resultados previos**

En [Hsu97] se formaliza el problema de optimizar la calidad del vídeo a transmitir seleccionando conjuntamente la tasa de compresión (el número de bits usados por una trama) y la tasa del canal (número de bits usados transmitidos para un intervalo de trama). En este trabajo se asume un retraso fijo de red. Para la optimización se usa un algoritmo Viterbi, que es muy costoso, lo que hace inviable su uso *on-line*.

En [Saha97] se obtienen los parámetros de un *leaky bucket* para un planificador PGPS (WFQ) que maximiza el número de conexiones admitidas dada unas trazas del tráfico y el tamaño del *buffer* B. Para ello, se formula el problema con programación lineal para obtener el valor óptimo de una forma parecida a [Reibman95].

En [Chen97] se utilizan multiplicadores de Lagrange para optimizar la codificación, siempre con un retraso de red fijo para tráfico CBR y VBR. También se presenta un método subóptimo con una complejidad reducida.

En [Luo97][Pancha94] se estudia cómo varía la calidad del vídeo variando la tasa de codificación con diferentes combinaciones de ancho de banda de red, tamaño del *buffer* y otros aspectos de codificación de MPEG-2. También se propone una codificación con dos prioridades, con resultados bastante pobres.

En [Rexford99] aunque no se utiliza *leaky bucket*, se desarrolla un método para minimizar el ancho de banda de red caracterizado con varias tasas de transmisión pico como función del retraso de comienzo de la transmisión y la memoria reservada en los nodos. Con esto, se presenta un algoritmo eficiente para minimizar el retraso de transmisión y la memoria reservada, sujeto a la restricción de una tasa de transmisión pico. Este algoritmo tiene coste  $O(n)$  donde  $n$  es el número de tramas. Con este retraso se suaviza el vídeo antes de transmitir.

Un esquema parecido es utilizado en [McManus96], donde dada la memoria del dispositivo de recepción (STB) B, se obtienen los pares óptimos de ancho de banda de un

canal CBR y el retraso en el comienzo de la transmisión  $d$  para evitar pérdidas de tramas durante la transmisión.

Las soluciones que se ofrecen en Internet se basan en adaptar la tasa de compresión al ancho de banda disponible en el momento. Si el ancho de banda disminuye durante la transmisión, el servidor (o el cliente) reducirá automáticamente la tasa de transmisión, reduciendo por tanto la calidad del vídeo o audio recibido.

#### 4.1.2.- *Objetivo de la optimización*

Cuando se van a reservar los recursos en la red para un determinado canal los parámetros de la red se conocen y son fijos. El problema reside, por tanto, en obtener los parámetros que caractericen el flujo y que optimicen la reserva de los recursos en la red. Como se verá en este capítulo este cálculo es iterativo y costoso y además se tiene que repetir para cada transmisión de un mismo vídeo. Es costoso porque implica recorrerse el vídeo para determinar los parámetros en cada iteración (por ejemplo, si se transmite una película se tendrá que recorrer 100 minutos de imagen que son unos 2Gbytes). En consecuencia, se propone un método alternativo basado en la envolvente empírica en el que se generan *off-line* una serie de puntos para cada vídeo que definen la carga. Utilizando estos puntos, se puede realizar *on-line* este mismo cálculo de una forma muy rápida y directa.

Otra de las conclusiones es que los parámetros de flujo que hacen óptima la reserva son iguales para las ecuaciones de retraso de Parekh/Galleger e IETF/RSVP usando planificadores del tipo WFQ. Esto implica una simplificación del cálculo para IETF/RSVP y una extensión de los resultados a ambos modelos.

#### 4.2.- *Cálculo del retraso emisor - receptor*

El retraso de emisor a receptor se calcula normalmente a partir de una descripción de la red y un modelo del tráfico, siendo normalmente función de la reserva realizada en los nodos. Como se ha visto en el capítulo 2 la especificación que hace IETF/RSVP de un servicio garantizado (*Guaranteed Service*) usa el modelo de *token bucket*  $(b,r,p)$  donde el número de bits que la fuente puede transmitir es menor que  $b+rt$  para cualquier intervalo de  $t^6$ . El retraso en cola se calcula por medio de las siguiente ecuaciones:

---

<sup>6</sup> Realmente si se establece una tasa pico  $p$ , el tráfico no puede exceder  $\min[pt, b+rt]$ .

$$d = \frac{(b-M)(p-R)}{R(p-r)} + \frac{M + C_{tot}}{R} + D_{tot} \quad (p > R \geq r) \quad (1)$$

$$d = \frac{M + C_{tot}}{R} + D_{tot} \quad (R \geq p \geq r) \quad (2)$$

Una ecuación más simple, que no utiliza la tasa pico para calcular el retraso, es la basada en el trabajo de Parekh y Gallager que es usada habitualmente para nodos con planificadores WFQ (otros planificadores también tienen esta ecuación como *Virtual Clock*, WF<sup>2</sup>Q, FFQ, SCFQ [Zhang98][Varma97]). Si el tráfico está limitado por un flujo *leaky bucket* ( $\mathbf{s}, \mathbf{r}$ )  $l(t) = \mathbf{s} + \mathbf{r}t$ , entonces el retraso máximo entre emisor y receptor para la sesión  $i$  es:

$$d_i = \frac{\mathbf{s} + nL_i}{R} + \sum_{j=1}^n \frac{L_{max_j}}{C_j} \quad (R \geq \mathbf{r}) = \frac{\mathbf{s} + C_{tot}}{R} + D_{tot} \quad (R \geq \mathbf{r}) \quad (3)$$

donde  $L_i$  es el tamaño máximo del paquete para la sesión  $i$ ,  $L_{max_j}$  es el tamaño máximo del paquete en el nodo  $j$ ,  $C_j$  el ancho de banda del nodo  $j$  y  $n$  el número de nodos. Esta ecuación es equivalente a (1) cuando la tasa pico  $p$  tiende a infinito (los parámetros  $r$  y  $b$  son equivalentes a los  $\mathbf{s}$  y  $\mathbf{r}$ ,  $L_i = M$  y  $L_{max_j} = MTU_j$ ). El tamaño del *buffer* en el nodo  $j$  será  $\mathbf{s} + jL_i$ .

En las ecuaciones de retraso descritas es importante fijarse que el retraso es inversamente proporcional a la reserva de ancho de banda a realizar. Esto implica, que para que una conexión obtenga un retraso pequeño tiene que reservar un ancho de banda elevado, incluso mayor que la tasa pico. Esto puede provocar que se desperdicien recursos cuando una conexión que necesita retrasos pequeños tenga poco tráfico.

Es importante remarcar que a las ecuaciones descritas (1)(2)(3) hay que añadir la latencia mínima de la red para calcular el retraso total. La latencia mínima es fija y dependerá principalmente de los medios de transmisión físicos, que normalmente suele ser despreciable respecto al retraso en red, aunque este valor puede ser importante en enlaces transoceánicos o vía satélite.

Como el objetivo es calcular la reserva a partir del retraso exigido, en las ecuaciones (1) y (3) se puede despejar  $R$  quedando:

$$R = \frac{(b-M)p + (M + C_{tot})(p-r)}{(d - D_{tot})(p-r) + b - M} \quad (p > R \geq r) \quad (4)$$

$$R = \frac{\mathbf{s} + C_{tot}}{d - D_{tot}} \quad (R \geq r) \quad (5)$$

En el caso que fuera necesario considerar la latencia mínima, se restará al retraso pedido esta latencia y se utilizarán las ecuaciones anteriores.

Resumiendo, para calcular la reserva de recursos se necesita: (1) Un modelo de la red, para obtener los parámetros  $C_{tot}$ ,  $D_{tot}$  y  $M$ , que ya está determinado, y (2) el cálculo de los parámetros del flujo  $(b, r, p)$  o  $(\mathbf{s}, \mathbf{r})$ . De estos parámetros, la tasa pico es de cálculo directo y determinado. El problema es cómo calcular los parámetros  $\mathbf{s}$  y  $\mathbf{r}$  para optimizar la reserva<sup>7</sup>.

### 4.3.- Optimización de la reserva de ancho de banda

Las secuencias de vídeo constan de  $n$  tramas, donde en cada trama  $i$  se transmiten una serie de bits  $E_i$ . La transmisión de vídeo está limitada por un conformador *leaky bucket*<sup>8</sup> donde no se puede transmitir  $\mathbf{s} + \mathbf{r}t$  bits para un tiempo  $t$ . Si la tasa de transmisión del cubo es  $\mathbf{r}$  y el número de tramas por segundo es  $f$  entonces el número de bits que tiene el cubo en la trama  $i$ , ignorando la capacidad finita del cubo es [Reibman95]:

$$\mathbf{s}_i = \max\left\{0, \mathbf{s}_{i-1} + E_i - \frac{\mathbf{r}}{f}\right\} \quad (6)$$

Por lo tanto, el tráfico será conforme si se escoge la capacidad del cubo  $\mathbf{s}$  tal que  $\mathbf{s}_i \leq \mathbf{s} \quad \forall i$ . Se puede expresar  $\mathbf{s}$  como función de  $\mathbf{r}$ :

$$\mathbf{s}(\mathbf{r}) = \max_{0 < i \leq n} \mathbf{s}_i \quad (7)$$

La figura 4.2 representa el tráfico denominado LAMBS1500<sup>9</sup>. En la figura 4.3 se muestran los valores de  $\mathbf{s}_i$  en función del tiempo para una tasa  $\mathbf{r}$  de 1Mb/s.

<sup>7</sup> A partir de aquí, se utilizará la nomenclatura  $\mathbf{s}, \mathbf{r}$  por ser la más usada para expresar tanto los parámetros *leaky bucket* como los parámetros  $b$  y  $r$  de *token bucket*.

<sup>8</sup> Esto es también aplicable al *token bucket*. El emisor está limitado por la misma ecuación, la diferencia reside en que se permite una transmisión a una tasa pico  $p$ .

<sup>9</sup> Los tráfico MPEG usados en esta tesis están descritos en el apéndice 1.

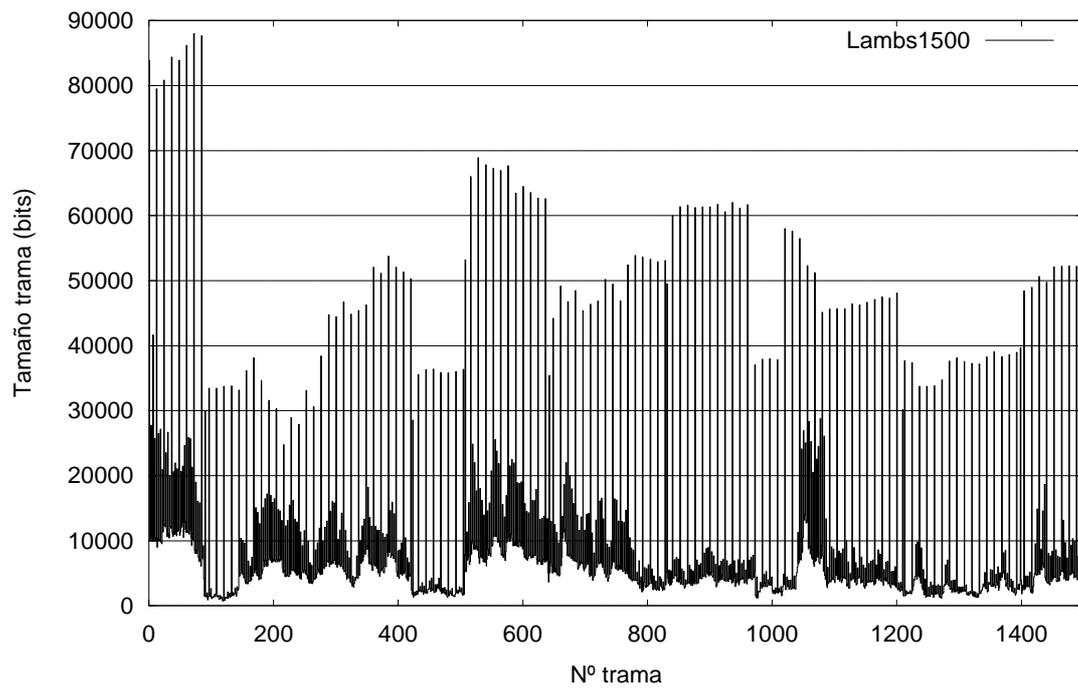


Figura 4.2: Tráfico LAMBS1500.

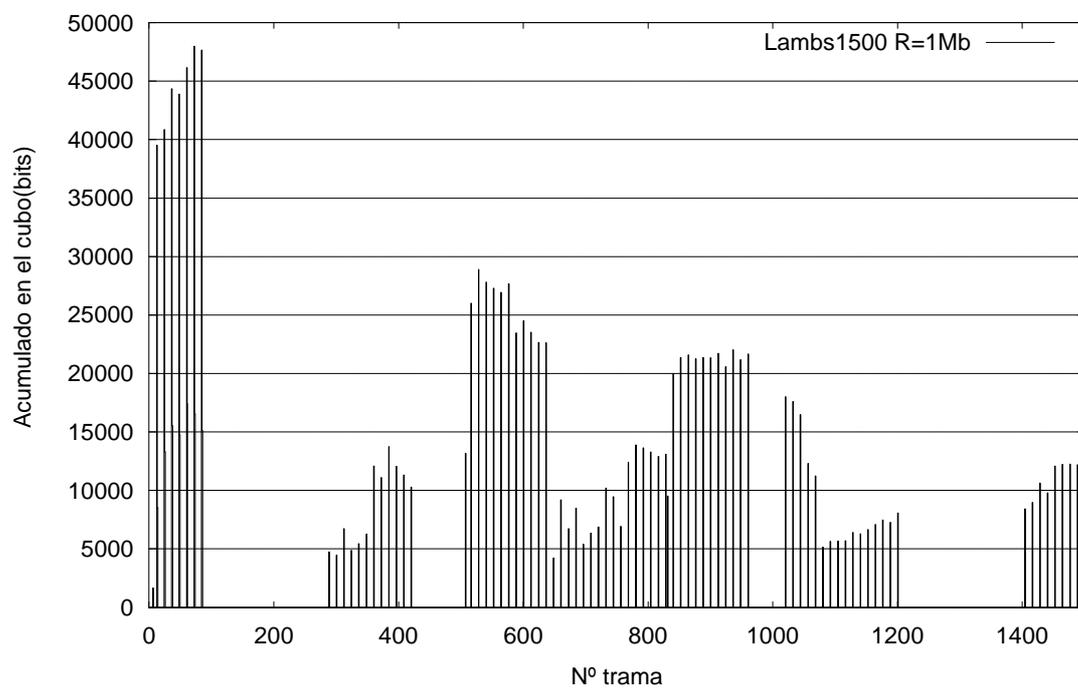


Figura 4.3: Valor  $\sigma$  en función del n° de trama para el tráfico LAMBS1500.

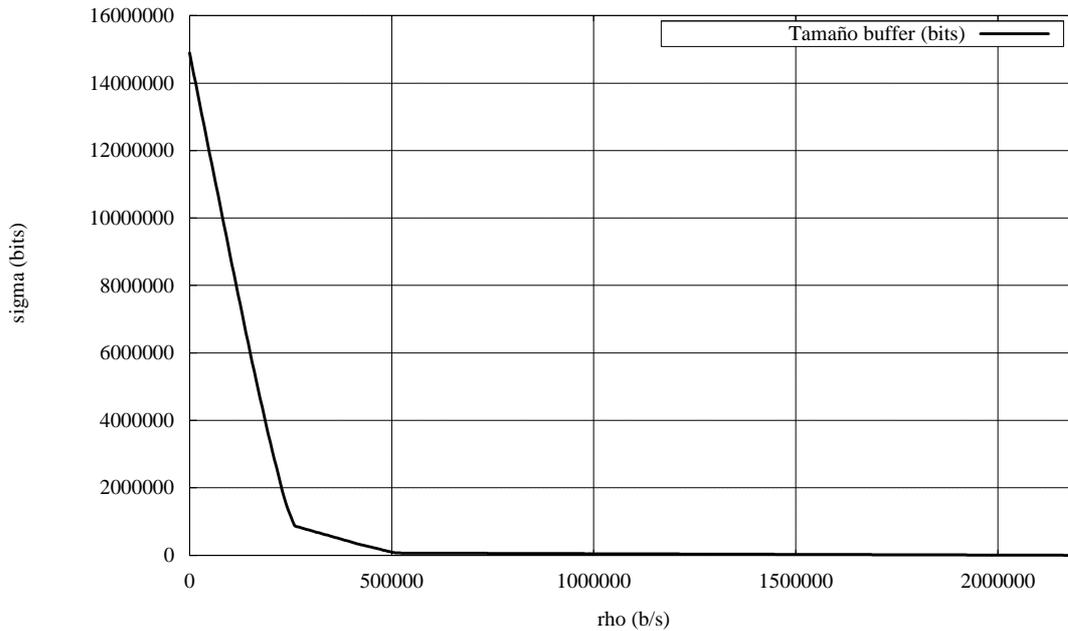
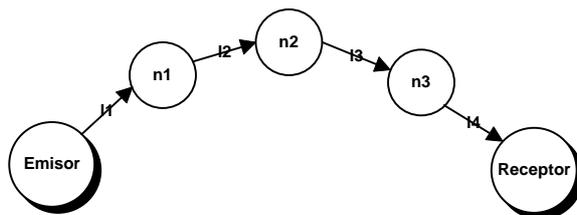


Figura 4.4: Función  $\mathbf{s}(\mathbf{r})$ : Profundidad de cubo necesario en función de la tasa de drenaje  $\mathbf{r}$ .

Se puede comprobar que distintos pares de valores  $\mathbf{s}, \mathbf{r}$  proporcionan un cálculo del retraso diferente. Usando, por ejemplo, la red de la figura 4.5, se tiene que los parámetros asociados a la red son  $D_{tot} = 0.0045s$  y  $C_{tot} = 40000b$ . Usando la ecuación (1) con el tráfico LAMBS1500 para un primer par de parámetros  $\mathbf{r}_1 = 240000b/s$ ,  $\mathbf{s}_1 = 1464008b$  la reserva para un retraso de 0.1s es de 2008639b/s. Para el par  $\mathbf{r}_2 = 590000b/s$ ,  $\mathbf{s}_2 = 64392b$  la reserva para el mismo retraso es de 961627b/s. Se ve que las reservas difieren mucho.



Enlace	AB (Mbps)	MTU (bits)
11	10	10000
12	15	12000
13	20	15000
14	10	20000
M = 10000		m = 1000

Figura 4.5: Red de ejemplo con tres nodos.

Por lo tanto, el objetivo es optimizar la reserva  $R$  en la red usando las ecuaciones (4) ó (5). Para ello se puede utilizar un método iterativo para encontrar los valores de  $\mathbf{r}$  y  $\mathbf{s}$  que optimicen la reserva tal como se sugiere en [Lakshman98]. El valor de  $\mathbf{r}$  puede variar entre 0 y  $p$ . El cálculo de  $\mathbf{s}$  depende del valor de  $\mathbf{r}$  y es una función decreciente. Aunque  $\mathbf{s}(\mathbf{r})$  en principio no es continua, debido a que el cálculo se realiza usando números enteros, para poder iterar se usará la misma ecuación (7) con todos sus parámetros reales devolviendo un valor  $\mathbf{s}$  real.

Sustituyendo en (4) y (5) los valores de  $r$  y  $b$  por  $\mathbf{r}$  y  $\mathbf{s}(\mathbf{r})$  quedan las siguientes funciones que calculan la reserva  $R$  en función de  $\mathbf{r}$ :

$$R(\mathbf{r}) = \frac{(\mathbf{s}(\mathbf{r}) - M)p + (M + C_{tot})(p - \mathbf{r})}{(d - D_{tot})(p - \mathbf{r}) + \mathbf{s}(\mathbf{r}) - M} \quad p > R(\mathbf{r}) \geq \mathbf{r} \quad (8)$$

$$R'(\mathbf{r}) = \frac{\mathbf{s}(\mathbf{r}) + C_{tot}}{d - D_{tot}} \quad R'(\mathbf{r}) \geq \mathbf{r} \quad (9)$$

A continuación se demostrará que la reserva óptima ocurre cuando  $R(\mathbf{r}) = \mathbf{r}$ , es decir, cuando la reserva es igual a la tasa de drenaje  $\mathbf{r}$ . Además, se demostrará que este valor  $\mathbf{r}$  minimiza  $R'(\mathbf{r})$  y que por tanto  $R(\mathbf{r}) = R'(\mathbf{r}) = \mathbf{r}$ . Esto simplifica, por tanto, la búsqueda de la reserva óptima al poder utilizar la ecuación (9), que es más simple, para ambos modelos.

**Lema 1:** La función  $R(\mathbf{r})$  es decreciente en el dominio  $[0, p]$  con la siguiente condición:  $(M + C_{tot}) > (d - D_{tot})$

**Demostración:** Dado que  $\mathbf{s}(\mathbf{r})$  es decreciente,  $\mathbf{s}(\mathbf{r}) - M$  es decreciente;  $p - \mathbf{r}$  es decreciente también. Por lo tanto, la función se puede expresar de la siguiente forma, donde  $\Delta_1(\mathbf{r}) = \mathbf{s}(\mathbf{r}) - M$  y  $\Delta_2(\mathbf{r}) = p - \mathbf{r}$  son funciones decrecientes:

$$R(\mathbf{r}) = \frac{p\Delta_1(\mathbf{r}) + (M + C_{tot})\Delta_2(\mathbf{r})}{\Delta_1(\mathbf{r}) + (d - D_{tot})\Delta_2(\mathbf{r})}$$

Una condición suficiente para que la función sea decreciente es que los términos que multiplican a  $\Delta_1(\mathbf{r})$  y  $\Delta_2(\mathbf{r})$  en el numerador sean mayores que en el denominador. Esto es:

- $p > I$ : evidentemente se trabaja con tasas pico mucho mayores que 1b/s, y
- $(M + C_{tot}) > (d - D_{tot})$ : esta es la condición que se ha impuesto. ¶

La condición impuesta se cumple normalmente:  $(M + C_{tot})$  es un múltiplo de  $M$ , que es el tamaño máximo de un paquete.  $M$  siempre será un entero positivo y bastante mayor que  $(d - D_{tot})$ , que como máximo será el retraso pedido en segundos y que normalmente es un valor reducido. Por ejemplo, en ATM el término  $(M + C_{tot})$  será como mínimo mayor que 428, lo cual da un margen para el retraso de 428s.

**Lema 2:** La función  $R'(r)$  es decreciente en el dominio  $[0, p]$  como lo es la función  $R(r)$ .

**Demostración:** Dado que  $s(r)$  es decreciente y el resto de los elementos son constantes,  $R'(r)$  es decreciente. ¶

**Teorema 1:** La reserva óptima ocurre cuando  $R(r) = r$ .

**Demostración:** La demostración es evidente. Dado que  $R(r)$  es decreciente en las condiciones indicadas en el lema 1 se escogerá el mínimo valor que cumpla la condición  $p > R(r) \geq r$ . El mínimo valor será por tanto  $R(r) = r$ . Este es, por tanto, el criterio de convergencia. ¶

Este criterio tiene una consecuencia muy importante:

**Corolario 1:** Los valores de  $r$  y  $s$  que hacen óptima la reserva en la ecuación (8) también lo son para la ecuación (9), es decir  $R'(r) = r$  cuando  $R(r) = r$ .

**Demostración:** Dado que la condición de convergencia es  $R(r) = r$  si se sustituye en la ecuación (8)  $r$  por  $R(r)$  se obtiene una ecuación de 2º grado. Resolviendo para  $R$  se tienen las siguientes dos soluciones:

- 1)  $R(r) = p$
- 2)  $R(r) = \frac{s(r) + C_{tot}}{d - D_{tot}}$

La primera solución no cumple la condición  $p > R(r)$ , pero la segunda es equivalente a la ecuación (9). Por lo tanto,  $R(r) = R'(r)$  cuando  $R(r) = r$ . ¶

Este corolario supone que con el criterio de convergencia establecido, la reserva con las dos ecuaciones es equivalente. Esto implica que la especificación más compleja del tráfico en RSVP/IETF (el parámetro  $p$ ) es innecesaria para obtener la reserva óptima en el caso de que todos los nodos utilicen planificadores del tipo WFQ. Dado que la ecuación de RSVP/IETF se basa en los parámetros  $C$  y  $D$  de los nodos, en el caso de que algún nodo no utilice este tipo de planificadores se seguiría utilizando la ecuación (8) para optimizar la reserva.

Esto permite extender la aplicación de este esquema a cualquier planificador que siga el modelo de retraso de la ecuación (3) y utilizarla para buscar la reserva óptima.

Otro tema importante es determinar para qué valores puede converger, es decir, cuándo se cumple la condición  $R'(\mathbf{r}) = \mathbf{r}$ :

**Corolario 2:**  $R'(\mathbf{r})$  converge a  $\mathbf{r}$  si  $\frac{C_{tot}}{d - D_{tot}} \leq p$ .

**Demostración:** Se tiene que el rango de  $\mathbf{r}$  es  $[0, p]$ . Si  $R'(\mathbf{r})$  converge a  $\mathbf{r}$  tendrá que tener algún valor en el rango  $[p, 0]$ . Como  $R'(\mathbf{r})$  es decreciente y depende únicamente de  $\mathbf{s}(\mathbf{r})$  que también es decreciente, el mínimo valor de  $R'(\mathbf{r})$  será cuando  $\mathbf{s}(\mathbf{r})$  es cero:

$\frac{C_{tot}}{d - D_{tot}}$ ; Para que haya al menos un valor en el intervalo  $[0, p]$  esto implicará que al menos  $\frac{C_{tot}}{d - D_{tot}}$  sea igual a  $p$ . ¶

Si no se cumple esta condición, para poder transmitir con el retraso requerido habrá que reservar más ancho de banda que el pico, con lo que se requiere utilizar una ecuación como la (2) que ya no depende del tipo de flujo.

Para encontrar las raíces de la ecuación (9) se puede utilizar cualquier método iterativo. Así, en el ejemplo del principio, los parámetros que hacen óptima la reserva para un retraso de 0.1s son  $\mathbf{r} = 944590\text{b/s}$  y  $\mathbf{s} = 50208\text{b}$ , con lo que la reserva es 944590b/s.

El problema es que este cálculo es muy costoso y no está acotado. En cada iteración se tiene que recorrer el vídeo para calcular  $\mathbf{s}$  en función de  $\mathbf{r}$ . Las pruebas realizadas con el tráfico LAMBS1500, la solución se encuentra entre 20 y 25 iteraciones dependiendo del retraso exigido, en la que en cada iteración hay recorrerse todo el vídeo. Se puede optimizar generando una traza del vídeo con el objeto de utilizarla para obtener  $\mathbf{s}$ . Pero el cálculo todavía es costoso y encima se tiene que almacenar un fichero grande con el tamaño de todas las tramas. Por lo tanto, se hace inviable el cálculo cada vez que se tiene que transmitir el vídeo.

Para solucionar este problema, se ha desarrollado un nuevo método basado en la envolvente empírica para que en el momento de la transmisión este cálculo sea rápido. Aunque el coste de generar la envolvente empírica es alto, tiene la ventaja de que se puede hacer *off-line* y se hace una sola vez para cada vídeo.

## 4.4.- Descripción de un método para la reserva óptima en-línea

### 4.4.1.- Introducción

Como se ha visto en el punto anterior es necesario obtener los valores  $\mathbf{s}, \mathbf{r}$  que optimicen la reserva de recursos, es decir, que optimicen la reserva en la ecuación (9). Para ello, en este punto se emplea la envolvente empírica con el objeto de obtener una información condensada, que permita optimizar la reserva. Esta información condensada se concreta en lo que se ha denominado puntos envolventes. A partir de los puntos envolventes se describirá un método rápido para obtener la reserva óptima.

Cualquier recta  $l(t)$  que contenga a la envolvente empírica estará definiendo los parámetros  $\mathbf{s}, \mathbf{r}$  del *leaky bucket* y por definición estará limitando el tráfico como se puede ver en la figura 4.6. Gráficamente, se puede ver, que para una misma envolvente empírica se pueden ajustar dos rectas  $l_1(t)$  y  $l_2(t)$ , lo que proporciona dos pares de valores  $\mathbf{s}$  y  $\mathbf{r}$ .

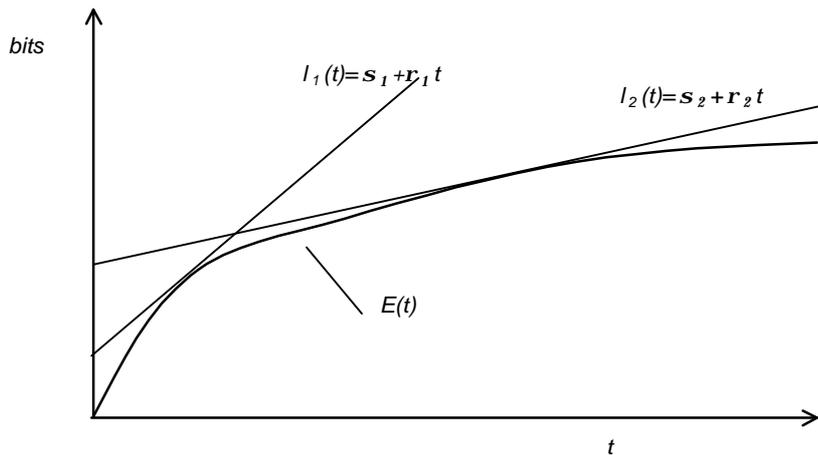


Figura 4.6: Ajuste a la envolvente empírica.

Como se ha visto en el punto anterior el valor de  $\mathbf{s}$  se calcula en función de la carga y de la tasa  $\mathbf{r}$ . Por ejemplo, si tenemos el siguiente tráfico  $\{ 1, 2, 0, 4, 2, 0, 1 \}$ , los valores de  $\mathbf{s}$  para los siguientes valores de  $\mathbf{r}$   $\{ 2, 1, 0.5, 0.25 \}$  son  $\{ 2, 4, 6.5, 8.25 \}$ . Para cada valor de  $\mathbf{r}$  se define por tanto una función de tráfico limitado  $l(t)$ . En la figura 4.7 se tiene la envolvente empírica sobre el tráfico y las rectas anteriores. Como se puede observar, las cuatro rectas tocan la envolvente empírica pero sin cortarla en ningún punto. Por la definición de la envolvente empírica toda recta  $l(t)$  que toque sin cortar será la mejor aproximación al tráfico y el valor en el origen será  $\mathbf{s}$ . El objetivo, por tanto, es obtener el conjunto de todas estas rectas. Viendo la figura se puede intuir como obtenerlas: las rectas

tocan a la envolvente empírica sólo en determinados puntos: el 1, 2, 5 y 7. La obtención de estos puntos servirá para calcular todas las rectas  $l(t)$ .

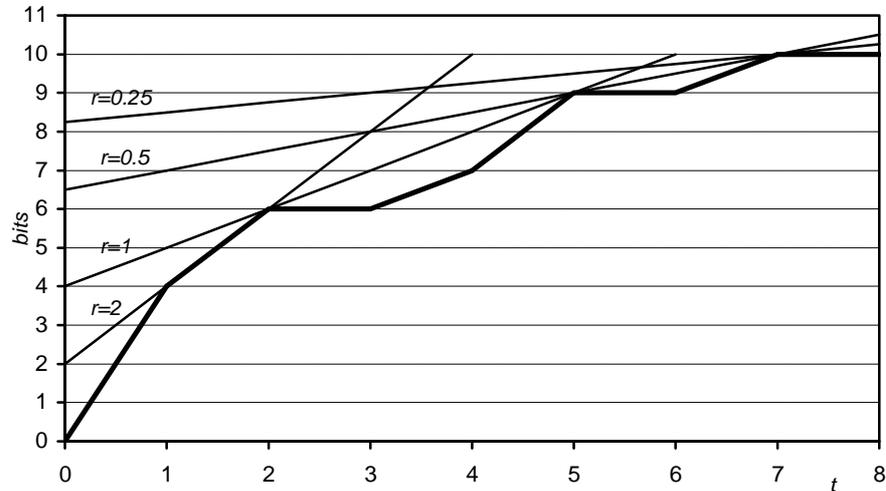


Figura 4.7: Envolvente empírica y alguna funciones  $l(t)$ .

#### 4.4.2.- Cálculo de los puntos envolventes

Como se intuye en el punto anterior la obtención de estos puntos permitirá calcular todas las posibles rectas  $l(t)$  más ajustadas a la envolvente empírica. Para ello, en este punto se introduce el concepto de puntos envolventes. Antes de desarrollar los puntos envolventes es necesario definir el concepto de “rectas envolventes” que es una formalización del conjunto de rectas más ajustadas o próximas a la envolvente empírica.

Si una función  $l(t) = s + rt$  describe el tráfico, cualquier función  $l(t)$  (se denomina al conjunto de estas rectas  $l^*(t)$ ) tendrá que contener  $E(t)$ . Esto es:

$$l^*(t) \geq E(t) \quad \forall t \geq 0 \quad (10)$$

**Definición:** Se llaman rectas envolventes al conjunto de rectas  $l(t)$  que contiene  $E(t)$  tal que no existe otra recta  $l'(t)$  con la misma pendiente  $r$  y  $l'(t) < l(t)$ .

Las rectas envolventes están definiendo en origen el valor  $s$  para la pendiente  $r$ . Es decir, a partir de las rectas envolventes se puede obtener el valor  $s$  en función de  $r$ , como se verá en este punto y el siguiente. Para obtener estas rectas envolventes se utilizarán los puntos envolventes.

Si  $E(t)$  fuese una función continua se podrían calcular las rectas tangentes para cualquier valor de  $t$  y tendríamos el conjunto de rectas  $l(t)$  más próximas. Dado que  $E(t)$  es

creciente, las pendientes en cualquier punto de la curva serían positivas<sup>10</sup>:  $E'(t) > 0 \forall t \geq 0$ , pero esto no significa que las pendientes sean siempre decrecientes. Esto implica que alguna de las rectas anteriores cortarían  $E(t)$  y no cumplen la condición (10). Para que las rectas tangentes a una función creciente no corten a ésta en un intervalo definido y estén siempre por encima, está función tendría que ser cóncava por debajo. Si  $E(t)$  cumpliera esta condición se obtendrían todas las rectas en función de  $t$ :

$$\mathbf{r}(t) = E'(t); \quad \mathbf{s}(t) = E(t) - \mathbf{r}t \Rightarrow \mathbf{s}(t) = E(t) - E'(t)t \quad (11)$$

Pero los tráficos reales son discretos, es decir, se tiene un conjunto de valores en función de unos incrementos de  $t$ , con lo que no se puede aplicar la derivada para hallar la tangente a la curva. La condición (10) se sigue cumpliendo. Por un punto  $t_i$  pueden pasar varias rectas con distintas pendientes  $\mathbf{r}$ , pero ninguna de estas rectas podrá cortar a  $E(t)$ . Sea  $l(t)$  la recta con menor pendiente que corta con un punto anterior de  $E(t)$  y  $l^+(t)$  la recta con mayor pendiente que corta con un punto posterior de  $E(t)$ . La condición para que  $t_i$  sea un punto envolvente sería que no exista otro punto anterior  $t_{i-1}$  donde la pendiente  $l(t_{i-1})$  sea menor a  $l(t_i)$  y ningún punto posterior  $t_{i+1}$  donde la pendiente  $l^+(t_{i+1})$  sea mayor a  $l^+(t_i)$ . En la figura 4.8a se pueden ver las dos rectas. Esto se puede formular por medio de la siguiente expresión:

$$l(t_i) = E(t_i) \Rightarrow \begin{cases} \mathbf{r}_i^- \geq \mathbf{r}^- & \forall t < t_i \quad l_i(t) = E(t) \\ \mathbf{r}_i^+ \leq \mathbf{r}^+ & \forall t > t_i \quad l_i(t) = E(t) \end{cases} \quad (12)$$

El objetivo consiste en obtener una serie de puntos de  $E(t)$  donde la pendiente de la recta que une la anterior sea siempre mayor que la pendiente de los puntos posteriores y que la pendiente de recta que une al punto posterior sea menor a las anteriores. Formalmente:

$$\forall t_i \quad \mathbf{r}_{i-1}^- > \mathbf{r}_i^- \quad \mathbf{r}_i^+ < \mathbf{r}_{i+1}^+$$

Como  $\mathbf{r}_{i-1}^+ = \mathbf{r}_i^-$ ,  $\mathbf{r}_i^+ = \mathbf{r}_{i+1}^-$ , la condición anterior se puede reducir a:

$$\forall t_i \quad \mathbf{r}_{i-1}^- > \mathbf{r}_i^- > \mathbf{r}_{i+1}^-$$

---

<sup>10</sup> En este caso la notación  $E'(t)$  representa la función derivada de  $E$  en  $t$ .

como se ve en la figura 4.8b. Es decir, se obtienen las rectas que unen los puntos de forma que las pendientes son siempre decrecientes. En resumen, lo que esto implica es la eliminación de los “valles” de la envolvente empírica, quedando los “picos” (figura 4.7).

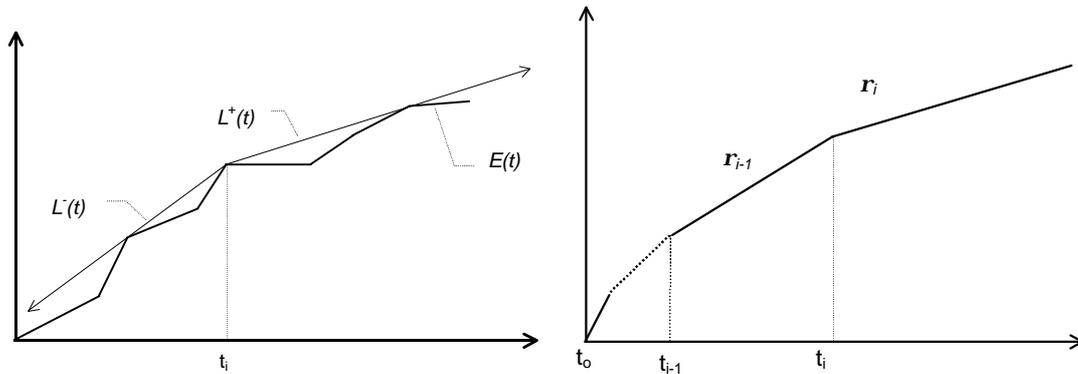


Figura 4.8: (a) Rectas  $L^-$  y  $L^+$ , (b) Pendiente decreciente para  $t_i$ .

<b>OBJETIVO</b> Calcula los puntos envolventes.
<b>ENTRADA :</b> $t_0, t_n$ : Intervalo de tiempo del tráfico $E(t)$ : Envolvente empírica.
<b>SALIDA:</b> $t_i, p_i, E_i$ : puntos, pendientes y valor de E en $t_i$ $m$ : Número de puntos generados
<b>ALGORITMO:</b> $t_0 = 0, r_0 = \frac{E(t_1) - E(t_0)}{t_1 - t_0} \text{ // Primer punto}$ $t_1 = 1;$ $i = 1;$ <b>para</b> $t$ <b>entre</b> $t_1, t_n$ $r = \frac{E(t) - E(t_{i-1})}{t - t_{i-1}}$ <b>mientras</b> $r \geq p_{i-1}$ // Se elimina el punto anterior $i = i - 1;$ $r = \frac{E(t) - E(t_{i-1})}{t - t_{i-1}} \text{ // Se vuelve a calcular la pendiente}$ <b>fmientras</b> <b>si</b> $r < p_{i-1}$ <b>entonces</b> // Genera un nuevo punto $t_i = t; r_{i-1} = r;$ $i = i + 1;$ $E_i = E(t);$ <b>fsi</b> <b>fpara</b> $m = i;$

Figura 4.9: Algoritmo para calcular los puntos envolventes.

Con esta condición se puede desarrollar un algoritmo para el cálculo de estos puntos que está detallado en la figura 4.9. La complejidad de este algoritmo es de  $O(n*m)$  donde  $n$  son los valores de  $t$  y  $m$  los puntos obtenidos. Este algoritmo, aunque diferente, produce el mismo resultado que el desarrollado en [Wrege96]. En este artículo se obtenían una serie de parámetros  $(\mathbf{s}, \mathbf{r})$  que especifican una función cóncava de  $n$  segmentos lineales que se utilizan para un planificador multi-tasa con un *leaky bucket* de  $n$  niveles.

#### 4.4.3.- Obtención de la reserva óptima

El objetivo de esta sección es demostrar que con los puntos envolventes se puede obtener una función  $\mathbf{s}'(\mathbf{r})$  equivalente a la función (7)  $\mathbf{s}(\mathbf{r})$ . Para ello, se demostrará que a partir de estos puntos se obtienen las rectas envolventes. A partir de esta demostración, se obtiene una simplificación a la hora de obtener la reserva óptima que permite una búsqueda de la solución rápida y acotada.

Para un punto  $i$  el valor de  $\mathbf{r}$  está en el intervalo  $[\mathbf{r}_{i-1}, \mathbf{r}_i]$ . Por este punto pasarán el siguiente conjunto de rectas:

$$E(t_i) = \mathbf{s} + \mathbf{r}t_i \quad \mathbf{r}_{i-1} \geq \mathbf{r} \geq \mathbf{r}_i \quad (13)$$

Despejando por tanto  $\mathbf{s}$  y variando  $\mathbf{r}$  en este intervalo se pueden calcular todas las rectas que pasan por el punto  $i$ :

$$\mathbf{s}_i(\mathbf{r}) = E(t_i) - t_i \mathbf{r} \quad \mathbf{r}_{i-1} \geq \mathbf{r} \geq \mathbf{r}_i \quad (14)$$

entonces  $\mathbf{s}'(\mathbf{r})$  es la composición de funciones lineales  $\mathbf{s}_i(\mathbf{r})$  para cada intervalo definido por los  $m$  puntos envolventes.

$$\mathbf{s}'(\mathbf{r}) = \mathbf{s}_i(\mathbf{r}) \quad \mathbf{r}_{i-1} \geq \mathbf{r} \geq \mathbf{r}_i \quad i = 1..m \quad (15)$$

es decir, que para calcular  $\mathbf{s}'(\mathbf{r})$  hay que buscar el intervalo donde esté comprendido  $\mathbf{r}$  y aplicar entonces la ecuación (15).

**Lema 3:**  $\mathbf{s}'(\mathbf{r})$  es monótona decreciente y continua en el  $i$ -intervalo  $[0, p]$ .

**Demostración**<sup>11</sup>: La función (14)  $\mathbf{s}_i(\mathbf{r})$  es monótona decreciente al depender sólo de  $\mathbf{r}$  en el intervalo  $[\mathbf{r}_{i-1}, \mathbf{r}_i]$ . El valor de  $E(t_0)$  es igual a cero con lo que  $\mathbf{s}(p)=0$ . Cuando  $\mathbf{r}$

---

<sup>11</sup> Hay que aclarar que los puntos que se generan tienen pendiente decreciente, es decir  $\mathbf{r}_{i-1} > \mathbf{r}_i$ . Esto significa que  $\mathbf{r}_i$  es igual a la tasa pico  $p$  y  $\mathbf{r}_m=0$ .

es cero el valor de  $s(0) = E(t_m)$ . Por definición, la recta que pasa por el punto  $t_i$  con pendiente  $r_{i-1}$  también pasa por el punto  $t_{i-1}$  y por lo tanto tiene el mismo valor en el origen  $s$ . Por ello  $s_i(r_{i-1}) = s_{i-1}(r_{i-1})$ . Para  $r_i$  es lo mismo y por tanto  $s_i(r_i) = s_{i+1}(r_i)$ . ¶

Además, por la propia definición,  $s_i(r)$  es una función decreciente por intervalos lineales como se puede ver en la figura 4.10. Se ha llegado por tanto, a un mecanismo por el cual se obtienen los segmentos comentados sobre la figura 4.4.

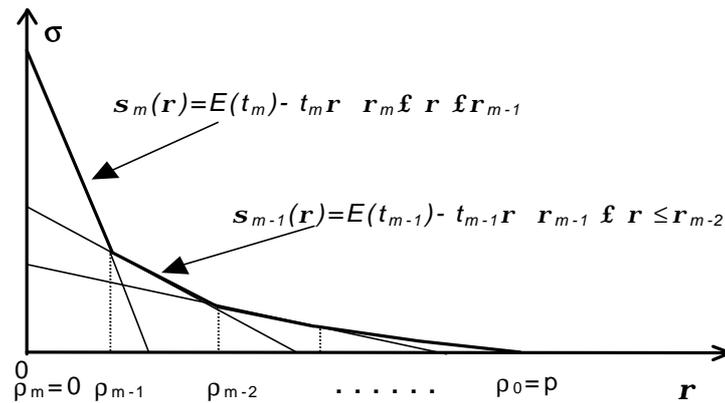


Figura 4.10: Función  $s'(r)$ .

Como se ve, el rango de  $s'(r)$  para el dominio  $[0, p]$  es  $[0, T]$  donde  $T$  es el total de bits transmitidos (que es igual a  $E(t_n)$ ). Se define  $L_r(t)$  como el conjunto de rectas  $l(t)$  generadas para cualquier valor de  $r$ . Esto es:

$$L_r(t) = \{ l(t) = s'(r) + rt \quad 0 \leq r \leq p \} \tag{16}$$

**Lema 4:**  $L_r(t)$  es el conjunto de rectas envolventes a la envolvente empírica  $E(t)$ .

**Demostración :** Si el rango de pendientes de  $L_r(t)$  va de 0 a  $p$ , lo que hay que probar es que no existe otra recta  $l'(t)$  con la misma pendiente tal que  $l'(t) < l(t)$ .

Por la condición (12) la recta  $l(t)$  tocará la envolvente empírica en un punto  $t_i$ . Si existiera una recta  $l'(t) < l(t)$  con la misma pendiente  $r$ , esto implicará que  $s' < s$ . Esto supone que el valor de  $l'(t)$  en  $t_i$  será menor que  $l(t_i)$ . Dado que  $l(t_i) = E(t_i)$ , esto significa que  $l'(t)$  corta a la envolvente empírica con lo que se incumple la condición (10).

Dado que  $s'(r)$  es continua y monótona decreciente (lema 3) se obtienen todas las rectas  $l(t)$  para cualquier valor de  $r$ . Por lo que  $L_r(t)$  es el conjunto de todas ellas. ¶

**Teorema 2:** La función  $\mathbf{s}'(\mathbf{r})$  es equivalente a  $\mathbf{s}(\mathbf{r})$ .

**Demostración:** Por el lema 4,  $L_r(t)$  es el conjunto de rectas envolventes de la envolvente empírica. Esto implica, que el valor en el origen  $\mathbf{s}$  de estas rectas es el menor para una tasa de drenaje  $\mathbf{r}$ , que es igual al valor  $\mathbf{s}(\mathbf{r})$ . Dado que  $\mathbf{s}'(\mathbf{r})$  es continua y decreciente entonces  $\mathbf{s}'(\mathbf{r}) = \mathbf{s}(\mathbf{r})$  para todo  $\mathbf{r}$  entre 0 y  $p$ . ¶

Por lo tanto, dado que  $\mathbf{s}'(\mathbf{r})$  devuelve los mismos valores que  $\mathbf{s}(\mathbf{r})$  se puede utilizar para iterar en la ecuación (9) y encontrar raíces de la misma forma. Esto permite obtener la solución a partir de los puntos envolventes, pero se sigue utilizando un método iterativo y no acotado para encontrar la solución  $R(\mathbf{r})=\mathbf{r}$ . Pero dado que  $\mathbf{s}'(\mathbf{r})$  está formada por segmentos lineales, esto permite desarrollar un método directo para encontrar la solución que se describe a continuación:

**Teorema 3:** Si la raíz  $R=\mathbf{r}$  está en el intervalo  $[\mathbf{r}_{i-1}, \mathbf{r}_i]$ , entonces:

$$R = \frac{E(t_i) + C_{tot}}{d - D_{tot} + t_i} \quad (17)$$

**Demostración:** Si la raíz  $R=\mathbf{r}$  está en  $[\mathbf{r}_{i-1}, \mathbf{r}_i]$ , se tiene que:

$$\mathbf{s}_i(\mathbf{r}) = E(t_i) - R t_i$$

y sustituyendo en la ecuación (9) y despejando  $R$ :

$$R = \frac{E(t_i) - R t_i + C_{tot}}{d - D_{tot}} \quad \Rightarrow \quad R = \frac{E(t_i) + C_{tot}}{d - D_{tot} + t_i} \quad ¶$$

Esto implica que si se sabe en qué intervalo está la raíz  $R=\mathbf{r}$  se puede obtener directamente el valor de  $R$ . La búsqueda del intervalo es fácil y acotado: se puede hacer una búsqueda binaria en los puntos envolventes hasta encontrar el intervalo. Cuando se encuentra el intervalo se aplicará el teorema 3 para obtener  $R$ . Un algoritmo para encontrar la solución se detalla en la figura 4.11. El coste de encontrar la solución es muy rápido, con un coste computacional de  $O(\log m)$ . Como se verá en el siguiente punto, el valor de  $m$  es pequeño (entre 20 y 100 puntos para los vídeos analizados).

La reserva mínima de *buffer* en los nodos puede ser calculada usando las ecuaciones vistas en el punto 4.2. El algoritmo descrito optimiza la reserva de ancho de banda dando un valor mínimo de  $\mathbf{s}$  para un retraso especificado. Si la restricción se impone

en el tamaño del *buffer*, esto implica una limitación en el valor de  $s$  y por tanto la reserva debe ser calculada usando las ecuaciones (4) y (5) directamente.

<b>OBJETIVO</b>
Calcula la reserva optimizada a partir de los puntos envolventes por intervalos
<b>ENTRADA :</b>
Parámetros de la red $C_{tot}$ y $D_{tot}$ . Puntos envolventes $t_i, p_i, E_i \ i = 0..n$
<b>SALIDA:</b>
$R, r, \sigma$ : Parámetros de reserva y flujo.
<b>ALGORITMO:</b>
$i_0 = 0; i_1 = n;$ // 1º Encuentra el segmento <b>hacer</b> $i = \frac{i_0 + i_1}{2};$ $s = S(r_i);$ $R = \frac{s + C_{tot}}{D - D_{tot}}$ <b>si</b> $R < r_i$ <b>entonces</b> $i_0 = i;$ <b>si</b> $R > r_i$ <b>entonces</b> $i_1 = i;$ <b>mientras</b> $(i_1 - i_0) > 1;$ // 2º Calcula el valor $R$ y $s$ $R = \frac{E_i + C_{tot}}{D - D_{tot} + t_i};$ $s = E_i - R * t_i;$ $r = R;$

Figura 4.11: Algoritmo para obtener la reserva óptima.

Resumiendo, el proceso completo está esquematizado en la figura 4.12 y se compone de dos fases diferenciadas. La primera fase, que es la más costosa, se puede realizar en el momento que se planifique introducir el vídeo en el servidor. La segunda fase ocurre cuando se establece un canal para la transmisión del vídeo. Detallando los procesos en cada fase sobre la figura 4.12:

- FASE I: El objetivo es calcular los puntos envolventes a partir del vídeo almacenado. Para ello, (a) se calcula la envolvente empírica a partir del vídeo, y (b) aplicando el algoritmo de la figura 4.9 se obtienen los puntos envolventes y se almacenan en un fichero que estará asociado al vídeo. El proceso más costoso de esta fase es el cálculo de la envolvente empírica que es de  $O(n^2)$ . Existen otros algoritmos más eficientes para calcular la envolvente empírica, pero dado que es un proceso *off-line* no es preocupante.
- FASE II: empieza cuando se establece un canal para la transmisión del vídeo. En esta fase un cliente (c) pide ver un vídeo iniciando el establecimiento de un canal

con una características determinadas, (d) en el servidor se calcula la reserva usando el algoritmo de la figura 4.11 sobre los puntos envoltantes calculados en la fase anterior y (e) se reservan los recursos calculados en la red empezando a continuación la transmisión. El coste de obtener la solución es, como se ha visto, de  $O(\log m)$ .

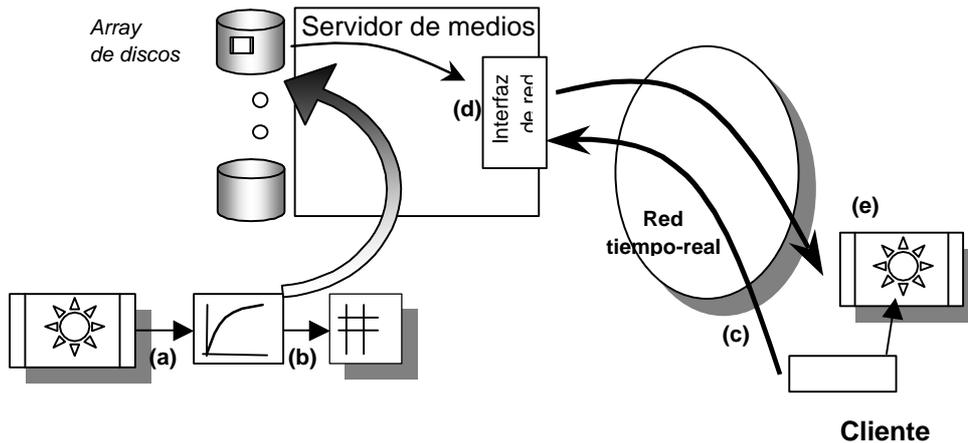


Figura 4.12: Proceso para obtener la reserva óptima.

#### 4.5.- Algunos resultados sobre tráfico MPEG

En este punto se muestran los resultados sobre algunas trazas de vídeos MPEG. Para ello se han implementado todos los algoritmos para obtener la reserva óptima. Por el corolario 1 las reservas son equivalentes para las ecuaciones del IETF/RSVP y Parekh. Por ejemplo, los puntos generados para el tráfico LAMBS1500 están en la figura 4.13. Como se puede ver, una de las propiedades más interesantes de estos puntos es que es una descripción sintetizada del tráfico. Así, el primer valor de  $r$  es siempre la tasa pico  $p$ .

Con estos datos y usando el ejemplo puesto en el punto 4.3 para un retraso máximo de 0.1s, se tiene que el par de valores obtenidos  $\sigma$  y  $r$  son 50208bits y 944590b/s, lo que conduce a una reserva de 944590b/s. Por el corolario 2 se puede calcular el retraso mínimo que se puede exigir que es  $D = \frac{C_{tot}}{p} + D_{tot} = 0.023s$ . En la figura 4.14 se muestra el cálculo

de la reserva y el valor  $\sigma$  en función del retraso exigido desde 0s a 5s en la red de ejemplo de la figura 4.5. Cuanto mayor es el retraso mayor tiene que ser el valor de  $\sigma$  debido a que se tiene que almacenar los tramas en la red. El incremento de este valor en función del tiempo es bastante lineal. En cambio, el valor  $p$  decrece exponencialmente hasta 0.25s para después decrecer linealmente.

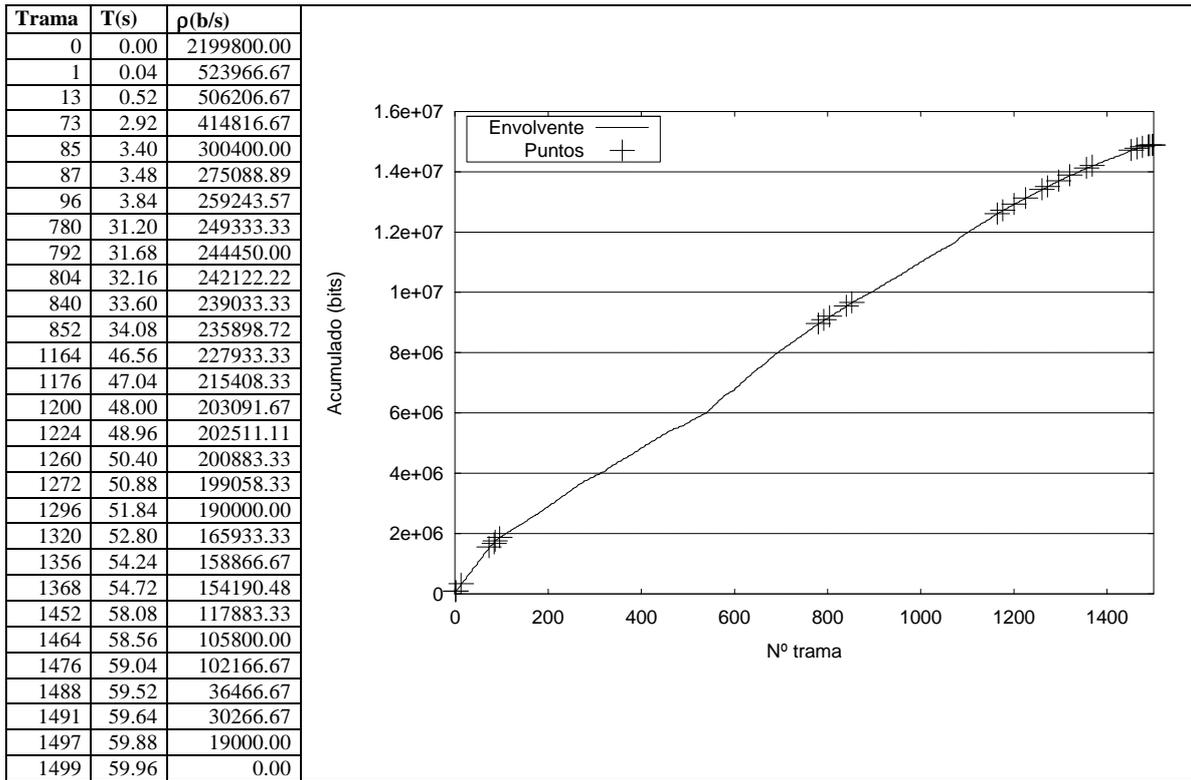


Figura 4.13: Puntos envolvente para el tráfico LAMBS1500.

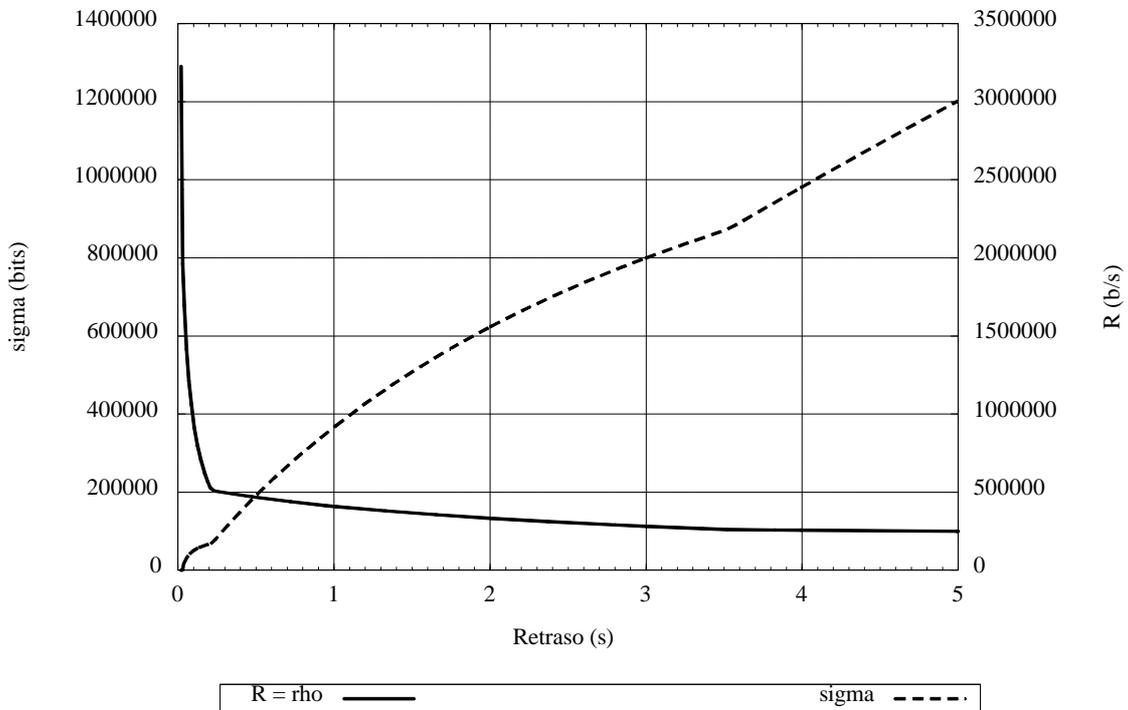


Figura 4.14: Retraso óptimo para tráfico LAMBS1500.

Se calcula también para el tráfico SOCCER completo que son 40000 tramas. En la figura 4.15 se muestran los puntos derivados. En este caso se han generado 37 puntos. El retraso mínimo que se puede exigir es 0.014s. En la figura 4.16 se obtiene la reserva óptima para retrasos entre 0s a 5s. En este caso, el punto de inflexión está sobre los 0.8s y es menos destacado.

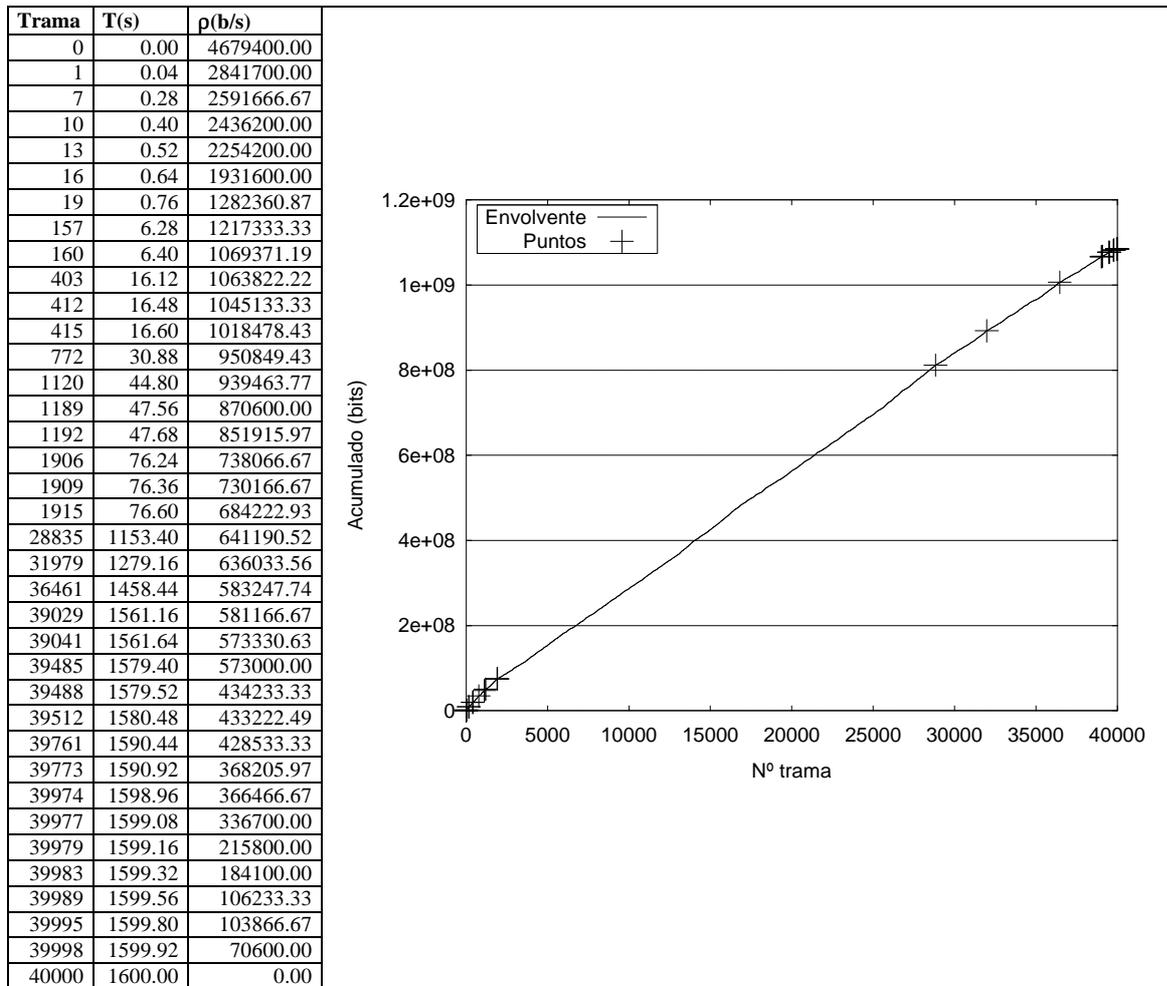


Figura 4.15: Puntos envolventes para el tráfico SOCCER.

En la tabla 4.1 se muestran el número de puntos generados usando las bien conocidas trazas de Rose [Rose95a]. Como se ve, el número de puntos está entre 34 y 84 puntos. En principio, no parece haber ninguna relación entre el número de puntos y la tasa media, pico, etc.

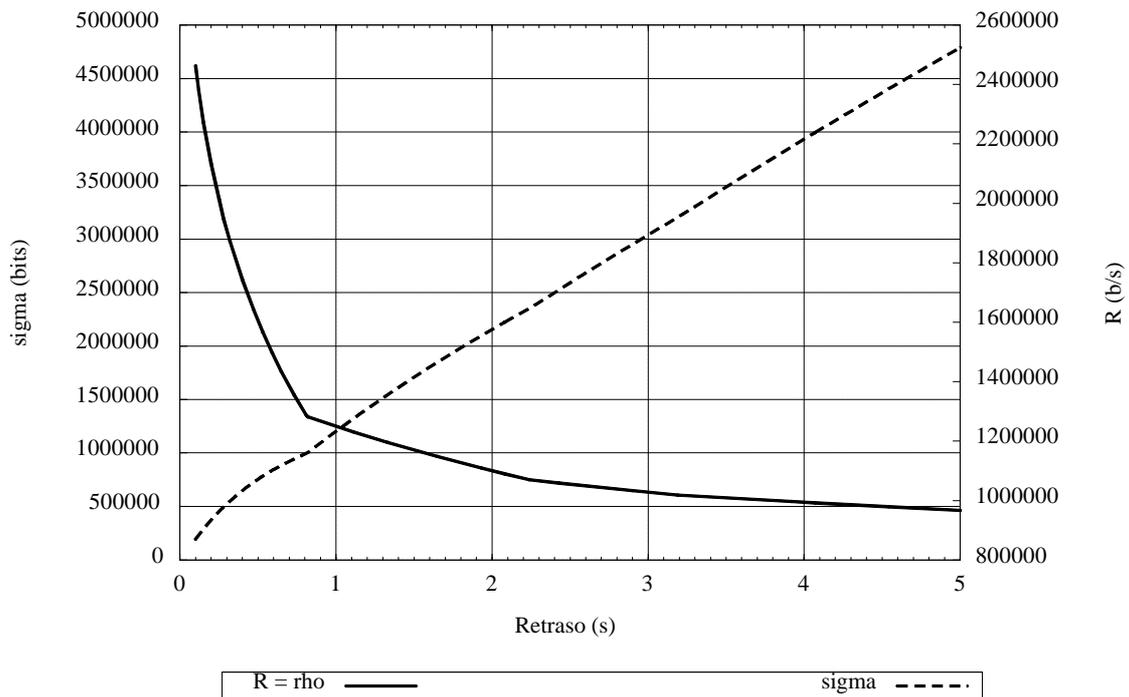


Figura 4.16: Retraso óptimo para el tráfico SOCCER.

Tráfico	Puntos	Tasa pico(b/s)	Media(b/s)	Pico/ Media
Asterix	70	3684400	558718	6.59
Atp	45	4771400	547246	8.72
bond	76	6114800	607711	10.10
Dino	66	2990800	326953	9.14
Lambs	65	3355600	182788	18.30
Movie	42	4316800	357208	12.08
MrBean	70	5726800	441177	13.00
Mtv1	56	5730000	615105	9.31
Mtv2	50	6285200	494512	12.71
News	34	1155000	150767	9.34
News2	59	4747200	383941	12.36
Race	35	5060400	768727	6.58
Sbowl	63	3521000	587649	5.99
Simpsons	45	6009400	464389	12.90
Soccer	37	4679400	678213	6.90
Soccer2	62	4757400	627739	7.58
Starwars	84	3120400	232829	13.40
Talk1	44	2669200	363420	7.34
Talk2	49	3318800	447874	7.41
Terminator	39	1989000	272618	7.29

Tabla 4.1: Número de puntos envolventes para los tráfico de Rose.

Sin embargo, siguiendo el ejemplo de la caracterización realizada en [Manzoni99] por tipo de traza, se puede ver que existe una ligera relación entre el número de puntos y el tipo de vídeo. En el rango de puntos envolventes entre 65 y 85 se tiene que todos son películas o series. En el rango 30 a 65 la mayoría son deportes (véase la figura 4.17). Estos

datos significativos, aunque no concluyentes, pueden dar pie a un profundo estudio estadístico de los puntos envolventes y su relación con otros modelos.

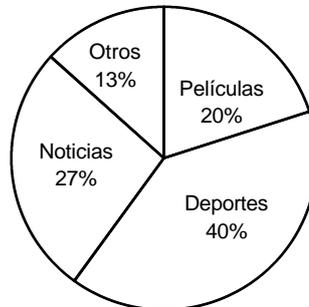


Figura 4.17: Tipos de tráfico en el rango 30 a 65.

#### 4.6.- Conclusiones

En este capítulo se ha introducido un nuevo método para obtener a partir de tráfico determinados los parámetros  $s$  y  $r$  que optimizan la reserva usando las ecuaciones desarrolladas por Parekh y las del servicio garantizado de RSVP/IETF. Este método, como se ha demostrado, obtiene la reserva óptima. Es aplicable a un gran número de planificadores que utilizan la misma ecuación del retraso (WFQ, *Virtual Clock*, WF<sup>2</sup>Q, FFQ, SCFQ).

Este método permite un cálculo muy rápido de los parámetros  $s$  y  $r$  en el momento de establecer el canal. El coste es  $O(\log m)$  donde  $m$  es pequeño (entre 34 y 84 puntos) frente al coste no acotado del método iterativo  $O(\log n * i)$  donde  $i$  son las iteraciones para encontrar la solución. También se ha propuesto el mecanismo de utilización dentro de la transmisión de vídeo bajo demanda.

Otra conclusión importante que se deriva de este trabajo es que los valores de  $s$  y  $r$  que optimizan la reserva y por tanto la reserva son iguales en la especificación de RSVP/IETF usando WFQ y la de Parekh/Gallagher. Esto significa, que la reserva óptima es equivalente cuando  $R=r$ , es decir, la reserva es equivalente a la tasa de drenaje.

# 5.

## Optimización de la memoria en los nodos

### 5.1.- Introducción

En el capítulo anterior, se ha visto un esquema para obtener la reserva óptima a partir de las características del tráfico y los datos de la red (es decir, de los nodos que componen el canal). En este capítulo se plantea otra optimización: la de minimizar la reserva de memoria total  $B$  (*buffer*) en un nodo que tiene un enlace de capacidad  $C$  que da servicio a  $n$  canales, sin que se produzcan pérdidas de información (*buffer overrun*), tal como se muestra en la figura 5.1.

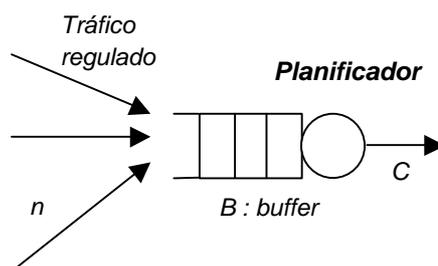


Figura 5.1: Nodo con planificador.

Algunas de las optimizaciones comentadas en la introducción del capítulo anterior tienen en cuenta la optimización de la memoria en los nodos. Pero este capítulo se va a centrar en mejorar las optimizaciones presentadas por A.Elwalid et al. [Elwalid95] y F.L.Presti et al.[Presti99].

En el artículo de Elwalid et al. se introduce un nuevo esquema para calcular la memoria en un nodo siempre que el tráfico esté conformado a la entrada con un modelo *token bucket*. Siempre que el tráfico siga un patrón ON-OFF se calcula la memoria necesaria tanto para un servicio sin pérdida como para uno estadístico. En el caso sin pérdidas este cálculo no es óptimo, ya que para simplificar asumen que la reserva de ancho de banda y memoria en los nodos tiene que ser directamente proporcional al ancho de banda y memoria total del nodo.

El artículo de Presti et al. obtiene un método óptimo para calcular la memoria total necesaria para un conjunto de canales bajo las mismas premisas que el artículo anterior: el tráfico es conformado a la entrada y el tráfico sigue un patrón ON-OFF que denominan tráfico *greedy*.

Los modelos de tráfico ON-OFF han sido ampliamente utilizados, sobre todo en el cálculo del ancho de banda efectivo para servicios estadísticos [Kvols92] [Roberts93].

En la primera parte de este capítulo, se introduce cómo calcular la memoria en los nodos a partir de tráficos conocidos y se describe con detalle la solución propuesta por Presti et al. A continuación, se introduce un nuevo método para optimizar esta memoria basado en los puntos envolventes usando la misma premisa que los dos artículos comentados: tráfico regulado a la entrada del nodo. Estos dos métodos se comparan usando tráficos MPEG.

La segunda parte de este capítulo describe cómo aplicar este esquema de optimización al caso de redes con varios nodos y subredes, introduciendo el concepto de macrocanal.

## 5.2.- Cálculo del uso de memoria en los nodos

Como se ha visto, se puede obtener el *buffer* necesario en función de una tasa de drenaje  $\mathbf{r}$  por medio de las ecuaciones (6) y (7) descritas en el capítulo anterior:

$$\mathbf{s}(\mathbf{r}) = \max_{0 < i \leq n} \mathbf{s}_i \quad \mathbf{s}_i = \max\{0, \mathbf{s}_{i-1} + E_i - \frac{\mathbf{r}}{f}\}$$

Como se ha demostrado, la función  $\mathbf{s}(\mathbf{r})$  es equivalente a  $\mathbf{s}'(\mathbf{r})$  que utiliza los puntos envolventes. Ambas funciones devuelven el *buffer* mínimo necesario para una tasa de drenaje determinada. En el caso de un nodo con varios canales, la memoria necesaria para  $m$  canales será la suma de los valores  $\mathbf{s}$  calculados para cada canal.

Este cálculo no tiene en cuenta la disposición temporal de los tráfico. Esto es, se está asumiendo la peor situación: todos los tráfico tienen su pico de transmisión en el mismo momento. Si se tiene en cuenta el tiempo y asumiendo que todos los tráfico tienen la misma base de tiempos, se puede obtener la memoria total necesaria en el nodo de la siguiente forma (donde  $\mathbf{s}_{i,j}$  es la función  $\mathbf{s}_i$  para el canal  $j$  y  $m$  el número de canales en el nodo):

$$\Omega_i = \sum_{j=1..m} \mathbf{s}_{i,j} \quad (1)$$

Esta expresión proporciona el uso de memoria para el tiempo asociado a la trama  $i$ . El tamaño máximo de memoria necesario será por tanto:

$$B = \max_{0 < i \leq n} \Omega_i \quad (2)$$

Como se ve, este cálculo necesita conocer de antemano las trazas de todos los tráfico que pasan por un nodo. Además, se necesita saber cuándo empieza cada vídeo. Estas dos consideraciones hacen que la ecuación (2) sólo se pueda usar en estudios analíticos. El valor de esta ecuación dependerá, por tanto, del tráfico que llega a los nodos y de la disposición temporal de ellos. Para simplificar, se trabaja con distintos modelos que permiten calcular el peor caso y que no dependen del tiempo.

El problema reside en pasar a los nodos información sobre la carga que se va a transmitir de tal forma que pueda ser útil para optimizar la reserva de recursos. Los puntos envolventes permiten que los nodos puedan usar la función  $\mathbf{s}'(\mathbf{r})$  para obtener el *buffer* necesario en función de la tasa de drenaje.

En el capítulo anterior, se ha visto que la optimización del ancho de banda se hace en el nodo emisor, habiendo obtenido de la red los parámetros que definen el canal que se va usar. En este caso el problema es diferente: el objetivo es minimizar en un nodo el *buffer* asignado a todos los canales. Para ello se juega con incrementar el ancho de banda asignado a un canal. Esto es, si un canal requiere como mínimo un ancho de banda  $\mathbf{r}$  esto implica una reserva de memoria  $\mathbf{s}$ . Si existe ancho de banda libre, se podría asignar mayor ancho de banda  $c$  a este canal para disminuir el *buffer* requerido, ya que como se ha visto un valor mayor de  $\mathbf{r}$  implica un valor menor de  $\mathbf{s}$ . El problema es cómo asignar el ancho de banda libre a los canales de tal forma que se reduzca la memoria total a reservar en el nodo. Se está por tanto, con un problema claro de optimización.

Como se comenta en el artículo de Presti, en este capítulo se trabaja únicamente con el tráfico conformado con el modelo *token bucket*. Como se vio en el capítulo 2, este

modelo permite inyectar en la red tasas de transmisión mayores que  $\mathbf{r}$ . En cambio, la utilización del modelo *leaky bucket* implica que el tráfico de entrada está limitado a  $\mathbf{r}$  con lo que no tiene sentido ampliar el ancho de banda asignado.

### 5.3.- Optimización de la reserva

En este punto se va a describir primero un método para optimizar la reserva propuesto por Presti et al. [Presti99] y luego se introducirá un nuevo método basado en los puntos envolventes.

Para optimizar el *buffer*, Presti parte únicamente de los parámetros del modelo *token bucket*  $(\mathbf{s}, \mathbf{r}, p)$ . Para ello, se asume que el tráfico es del tipo “*greedy*”. Se define el periodo  $T_{on}$  como el tiempo máximo de una ráfaga con la transmisión a tasa pico  $p$ :

$$T_{on} = \frac{\mathbf{s}}{p - \mathbf{r}} \quad (3)$$

Si una fuente genera tráfico a la tasa pico  $p$  para el intervalo de tiempo  $T_{on}$  y pasa a tasa  $\mathbf{r}$  para el resto de tiempo, entonces el tráfico se denomina *greedy* (véase la figura 5.2). Aunque este modelo es ampliamente utilizado por su sencillez es una aproximación muy basta. Suponiendo que hay  $J$  clases de tráfico, donde hay  $K_j$  canales para cada clase  $j$  tal que  $\sum_{j=1}^J K_j = n$  y donde todos los canales de una clase  $j$  tienen los mismos parámetros *token bucket*  $(\mathbf{s}_j, \mathbf{r}_j, p_j)$ . El objetivo es, por tanto, obtener los anchos de banda  $c_j$  asignados a cada clase tal que  $\sum_{j=1}^J K_j c_j \leq C$  y se minimice el *buffer* requerido en el nodo. Con esta información obtienen por programación lineal la solución óptima, ordenando las clases para que  $T_{on_1} > T_{on_2} > \dots > T_{on_J}$ .

Formalmente, sea  $k$  el menor índice tal que:

$$\sum_{j=1}^{k-1} K_j p_j + \sum_{j=k}^J K_j \mathbf{r}_j \leq C \leq \sum_{j=1}^k K_j p_j + \sum_{j=k+1}^J K_j \mathbf{r}_j \quad (4)$$

La forma de asignar los valores  $c_j$  para minimizar el *buffer* total es el que sigue:

$$c_j = \begin{cases} p_j, & j = 1, \dots, k-1, \\ \frac{C - \sum_{l \neq k} K_l c_l}{K_j}, & j = k, \\ r_j, & j = k+1, \dots, J \end{cases} \quad (5)$$

y el *buffer* asignado a cada canal será:

$$b_j = \begin{cases} 0, & j = 1, \dots, k-1, \\ s_j - T_{on_j}(c_j - r_j), & j = k, \\ s_j, & j = k+1, \dots, J \end{cases} \quad (6)$$

La suma de los valores  $b_j$  es el *buffer*  $B$  necesario en el nodo en función de  $C$ . El *buffer* necesario  $B$  es una función de  $C$  que es lineal por segmentos, decreciente y convexa [Presti99]. El número de segmentos es igual al número de clases  $J$  (véase la figura 5.3).

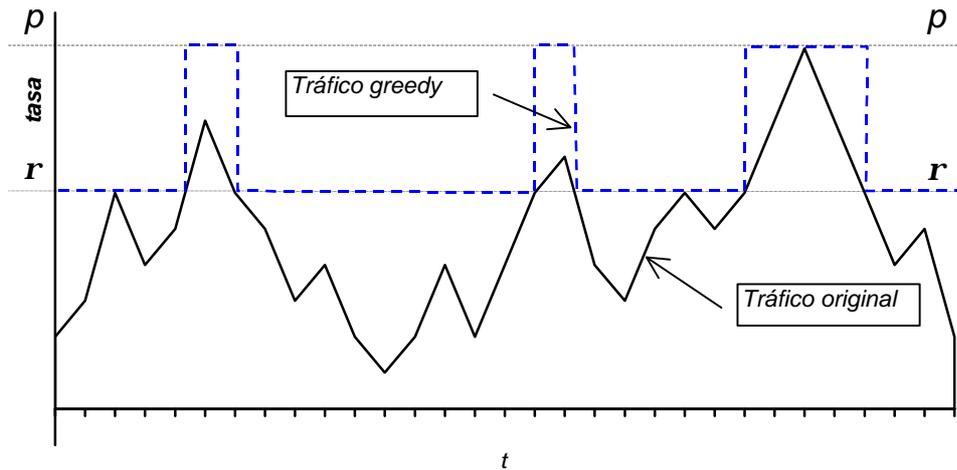


Figura 5.2: Tráfico greedy.

Tal como se ve en la figura 5.2, el tráfico tipo *greedy* es una cota muy alta del tráfico. Esta aproximación surge de la imposibilidad de obtener el valor de  $s$  a partir de  $r$  en los nodos. Con los puntos envolventes se dispone, por tanto, de un mecanismo que permite obtener los valores de  $s$  en función de  $r$  con muy poca información. Sin los puntos envolventes, esta información sólo se podría obtener recorriendo todo el tráfico a transmitir (lo que hace inviable que esta información la tengan los nodos). En cambio, los puntos envolventes se pueden transmitir a los nodos debido a su reducida cantidad (menos de cien puntos). De esta forma se permite a los nodos calcular de forma exacta (y rápida) el valor de  $s$ .

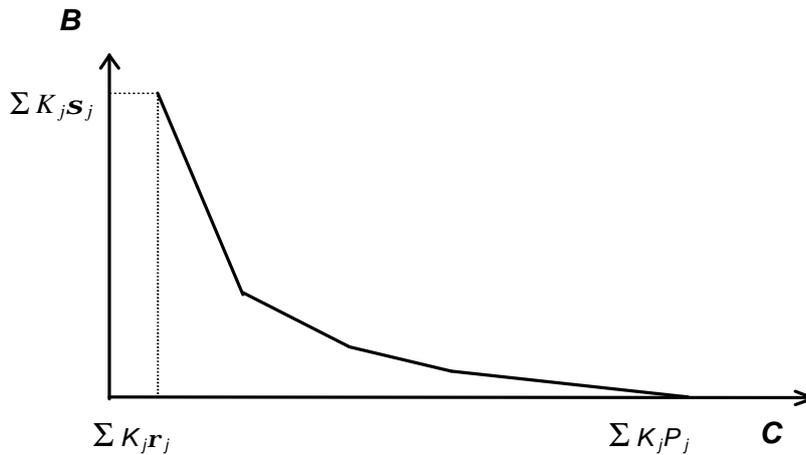


Figura 5.3: Función  $B=f(C)$ .

Dado que ya se puede calcular el valor de  $\mathbf{s}$  a partir de los puntos envolventes usando la función  $\mathbf{s}'(\mathbf{r})$ , el objetivo será, por tanto, minimizar la suma para todos los canales<sup>12</sup> (donde  $\mathbf{s}'_j(\mathbf{r})$  es la función  $\mathbf{s}'(\mathbf{r})$  para el canal  $j$ ):

$$\sum_{j=1}^n \mathbf{s}'_j(c_j) \quad \sum_{j=1}^n c_j = C \wedge c_j \geq \mathbf{r}_j \quad (7)$$

es decir, hay que encontrar los valores  $c_j$  que minimicen la suma total de  $\mathbf{s}_j$ . Dado que la función  $\mathbf{s}'_j(\mathbf{r})$  devuelve el valor  $\mathbf{s}$  mínimo para el canal  $j$ , la función  $\mathbf{s}'_j(c_j)$  devuelve el mínimo valor de  $\mathbf{s}$  para un nuevo ancho de banda  $c_j$ . El ancho de banda mínimo será aquella combinación de  $c_0, \dots, c_n$  que minimiza la suma de todas las funciones  $\mathbf{s}'_j(c_j)$  para todos los canales.

En el siguiente punto, se describe un método rápido para encontrar la combinación de  $c_0, \dots, c_n$  que minimiza la expresión (7).

#### 5.4.- Método para la optimización de la memoria en un nodo

En este punto, se introduce un método para minimizar el *buffer* usando la función  $\mathbf{s}'(\mathbf{r})$ . Como se ha visto, el problema reside en minimizar el sumatorio (7). Si cada canal tiene una tasa  $\mathbf{r}_j$  tal que:

<sup>12</sup> A partir de este momento ya no se trabaja con clases, todos los canales se tratan por igual aunque tengan los mismos parámetros.

$$\sum_{j=1}^n r_j \leq C \quad (8)$$

esto implica que existe un ancho de banda remanente  $C_R$  que se puede asignar a los canales  $c_j$ . Por lo tanto, el problema cómo asignar ese ancho de banda remanente a los distintos canales para minimizar el valor de la ecuación (7).

Como se vio en el capítulo anterior, la función  $\mathbf{s}'(\mathbf{r})$  es la composición de las funciones lineales definidas para cada intervalo por los puntos envolventes:

$$\mathbf{s}'(\mathbf{r}) = \mathbf{s}_i(\mathbf{r}) \quad r_{i-1} \geq \mathbf{r} \geq r_i \quad i = 1 \dots m$$

donde:

$$\mathbf{s}_i(\mathbf{r}) = E(t_i) - t_i \mathbf{r}$$

Estas rectas se pueden ver gráficamente en la figura 4.10. Cada recta tendrá una pendiente que es  $t_i$ .

Supongamos que todas las funciones  $\mathbf{s}_j'(\mathbf{r})$  sólo tienen un intervalo, con lo que cada función sería una recta de la forma  $E(t) - t\mathbf{r}$ . La pendiente vendría determinada por tanto, por el valor de  $t$ .

El incremento de ancho de banda  $c_j$  asignado a los canales a partir del valor inicial  $r_j$  implica una disminución de  $\mathbf{s}_j$  (las funciones  $\mathbf{s}_j'(\mathbf{r})$  son rectas de pendiente negativa  $t$ ). La mayor reducción se producirá en aquel canal con mayor pendiente  $t$ , con lo que para minimizar el total se cogerá aquel canal con el mayor valor de  $t$ . A este canal se le asignará un valor  $c_j$  que será la tasa pico  $p_j$  si la diferencia entre  $p_j$  y  $r_j$  es menor que el ancho de banda remanente  $C_R$  o sino  $C_R + r_j$ . En el caso que quedase ancho de banda, se buscaría el siguiente canal con mayor pendiente y se seguiría el mismo procedimiento hasta agotar el valor  $C_R$ . El algoritmo está en la figura 5.4. Primero se obtiene el valor  $C_R$  que es el ancho de banda disponible hasta el máximo asignable (que es cuando se llega a las tasas pico de los canales). Luego se realiza un bucle mientras quede ancho de banda disponible  $C_R$ , donde en cada iteración se busca el canal con la mayor pendiente entre los que no han llegado a su tasa pico ( $c_j \neq p_j$ ).

<b>OBJETIVO</b> Optimizar <i>buffer</i> con un solo intervalo por canal
<b>ENTRADA :</b> C : Ancho de banda a asignar $r_0.. r_n$ : Tasas de drenaje para cada canal. $t_0.. t_n$ : Pendientes de la rectas. $R_0.. R_n$ : Reserva mínima para cada canal. $p_0.. p_n$ : Tasa pico de cada canal
<b>SALIDA:</b> $c_0.. c_n$ : Nueva reserva para cada canal.
<b>ALGORITMO:</b>  $C_R = \min(C - \sum_{j=1}^n R_j, \sum_{j=1}^n p_j - \sum_{j=1}^n R_j)$ $c_j = R_j \quad \forall j$ <b>mientras</b> $C_R > 0$ $j = \{ j \mid \max(t_j) \wedge c_j \neq p_j \}$ // Busca el índice de la máx. pendiente. <b>si</b> $p_j - r_j \leq C_R$ <b>entonces</b> $c_j = p_j$ $C_R = C_R - (p_j - r_j)$ <b>sino</b> $c_j = r_j + C_R$ $C_R = 0$ <b>fsi</b> <b>fmientras</b>

Figura 5.4: Algoritmo optimización *buffer* (1 solo intervalo).

Este procedimiento se puede ampliar al caso en que  $\mathbf{s}_j'(\mathbf{r})$  esté compuesta por varios intervalos. Esto es,  $\mathbf{s}'(\mathbf{r})$  está definida como una serie de segmentos para cada intervalo. Cada canal  $j$  tendrá un valor  $r_j$  que define el intervalo  $i$ , y por lo tanto el segmento  $\mathbf{s}_i(\mathbf{r}) = E(t_i) - t_i \mathbf{r}$  con pendiente  $t_i$ . En consecuencia, se buscará el canal  $j$  cuya pendiente sea mayor y se incrementará el valor  $c_j$  hasta el menor de estos valores: el valor  $C_R$  o bien el valor  $r$  que define el intervalo anterior. Como se ve, la diferencia con el método anterior es que se tiene en cuenta el intervalo. Si queda ancho de banda por asignar, en el canal  $j$  se indica que está en el siguiente intervalo y se buscará la recta con mayor pendiente. El algoritmo está en la figura 5.5. Este algoritmo añade al anterior, el vector  $I$  donde se almacena la posición dentro de la matriz de puntos envolventes de cada canal, es decir, está definiendo el intervalo. Esto permite obtener la máxima pendiente en cada intervalo e ir pasando a un nuevo intervalo.

El coste computacional del algoritmo vendrá dado por la búsqueda del índice de la pendiente máxima  $t$  que puede ser  $O(\log n)$  y la asignación de recursos que dependerá del valor de  $C_R$  y el tamaño de los intervalos, que en principio no se puede acotar fácilmente. La memoria total  $B$  necesaria se obtendrá a partir del sumatorio de la ecuación (7).

<b>OBJETIVO</b> Optimizar <i>buffer</i> usando puntos envolventes
<b>ENTRADA :</b> C : Ancho de banda a asignar. P : { $r_{i,j}$ } Matriz de tasas de drenaje de los puntos envolventes. (Primer índice $i$ : n° de punto, segundo $j$ : número de canal) T : { $t_{i,j}$ } : Matriz de pendientes de las rectas. $R_0.. R_n$ : Reserva mínima para cada canal. $p_0.. p_n$ : Tasa pico de cada canal
<b>SALIDA:</b> $c_0.. c_n$ : Nueva reserva para cada canal.
<b>ALGORITMO:</b> $C_R = \min(C - \sum_{j=1}^n R_j, \sum_{j=1}^n p_j - \sum_{j=1}^n R_j)$ $c_j = R_j \quad \forall j$ $I_j = \{ \min(i) \mid r_{i,j} > R_j \} \quad \forall j // \text{ Busca posición índice tasa } i$ <b>mientras</b> $C_R > 0$ $j = \{ j \mid \max(t_{1..j}) \wedge c_j \neq p_j \} // \text{ Busca el índice de la máx. pendiente.}$ <b>si</b> $r_{i-1,j} - c_j \leq C_R$ <b>entonces</b> $C_R = C_R - (r_{i-1,j} - c_j)$ $c_j = r_{i-1,j}$ <b>sino</b> $c_j = r_{i,j} + C_R$ $C_R = 0$ <b>fsi</b> $I_j := I_j - 1$ <b>fmientras</b>

Figura 5.5: Algoritmo de optimización con los puntos envolventes.

## 5.5.- Implementación

Para que los nodos puedan asignar los anchos de banda  $c_j$  de forma óptima necesitan disponer de los puntos envolventes en el momento de crear el canal. Para ello, en el momento de establecer un canal se enviará esta información. Como se ha visto, lo que permite este mecanismo es que los puntos envolventes son pocos comparados con la traza total. Suponiendo un máximo de 100 puntos, esto supondrá 400 bytes de transmisión (4 bytes para representación en coma flotante). Además, no es necesario enviar todos los puntos ya que sólo interesan los puntos por encima de  $r_j$  con lo que se puede reducir el número de puntos a transmitir (y almacenar en los nodos) <sup>13</sup>.

<sup>13</sup> Esto se puede ver en el algoritmo de la figura 5.5 donde se van cogiendo los intervalos interiores que suponen valores superiores de  $r$ .

Dada la eficiencia del algoritmo para asignar los anchos de banda  $c_j$ , estos valores se pueden calcular dinámicamente cada vez que se crea o elimina un canal en el nodo. De esta forma, se puede calcular el tamaño del *buffer* a usar en cada intervalo.

La cota superior para obtener la asignación de ancho de banda para  $n$  canales con  $m$  puntos sería el coste de ordenar los  $n$  canales y valores  $t_i$ , es decir  $O(nm \cdot \log(nm))$ . De todas formas, el coste dependerá realmente del ancho de banda que sobra y la cantidad de canales que se puede asignar. Por ello, no parece conveniente ordenar la lista.

Para que el cálculo del *buffer* total necesario sea rápido se puede usar el siguiente esquema. Se crea en el nodo una tabla en la que se guardan todos los intervalos de los canales con las siguientes columnas:

- **n**: número de fila.
- $t_{i,j}$ : pendiente de la recta ( $t$ ) para el canal  $j$  en el intervalo  $i$ .
- **canal**: número de canal al que pertenece la fila.
- $c_{i,jmax}$ : máxima asignación posible de ancho de banda en el intervalo. Este valor es la diferencia para el intervalo  $i$  entre  $r_i$  y  $r_{i-1}$ , y define, por tanto, cuánto ancho de banda se asignará en ese intervalo.
- $c_{i,j}$ : asignación actual de ancho de banda para el intervalo. El valor será 0 si no está asignado o menor o igual que  $c_{i,jmax}$  si se ha asignado ancho de banda.

La tabla estará ordenada por orden decreciente de  $t_{i,j}$ . Además, se guarda el ancho de banda restante  $C_R$  y la posición del último canal asignado. Para facilitar la explicación, esta posición se denominará listón y estará representada por una raya negra gruesa. Un ejemplo de la situación de esa tabla es la mostrada en la tabla 5.1. En este ejemplo se tienen 3 canales donde el canal 1 tiene dos intervalos con pendientes  $t_i$  de 15 y 7. El canal 2 tiene 3 intervalos y el canal 3 tiene 2. La situación de partida es que hay 4 intervalos asignados (del 0 al 3) y  $C_R$  es cero. El último intervalo (el 3) sólo tiene asignado 2 unidades de ancho de banda  $c_{i,j}=2$  de los 4 posibles.

n	$t_{i,j}$	canal	$c_{i,j}$	$c_{i,j \max}$
0	15	1	5	5
1	12	2	3	3
2	10	3	4	4
3	7	1	2	4
4	6	2	0	5
5	4	3	0	6
6	2	2	0	2

Tabla 5.1: Situación inicial de la tabla.

Cuando se añade un nuevo canal en el nodo, se insertan los intervalos del canal en la lista manteniendo el orden de pendientes decrecientes  $t$ . Si alguno de los intervalos se ha insertado antes del listón se resta a  $C_R$  el valor  $c_{i,j}$ . Si después de la inserción el valor de  $C_R$  es negativo se eliminarán los últimos intervalos por encima del listón sumando el valor  $c_{i,j}$  hasta que  $C_R$  sea 0. Siguiendo con el ejemplo, si entra un nuevo canal 4 con 3 intervalos  $(t_i, c_{i \max}) \{ (13,2), (8,4), (4,5) \}$  esto implica que se inserta el primer intervalo del canal 4 en la 2ª posición (encima del listón) con lo que el valor de  $C_R$  será  $-2$ . Esto implicará que se eliminará al canal 1 las 2 unidades asignadas (se sube una posición el listón). La situación queda tal como refleja la tabla 5.2.

n	$t_{i,j}$	canal	$c_{i,j}$	$c_{i,j \max}$
0	15	1	5	5
1	13	4	2	2
2	12	2	3	3
3	10	3	4	4
4	8	4	0	4
5	7	1	0	4
6	6	2	0	5
7	4	3	0	6
8	4	4	0	5
9	2	2	0	2

Tabla 5.2: Se añade el canal 4.

Cuando se elimina un canal, se borrarán de la lista todas las entradas correspondientes al canal. Si el canal tenía algún intervalo con ancho de banda asignado (es decir, está por encima del listón) se sumará a  $C_R$ . Después de esto, si  $C_R$  es positivo se cogen los intervalos hasta que  $C_R$  sea 0 (se baja el listón). Siguiendo el ejemplo anterior, si se elimina el canal 2, tendremos que  $C_R$  será igual a 3 y podremos asignar al canal 4 un  $c_i$  de 3.

n	t <sub>i,j</sub>	canal	c <sub>i,j</sub>	c <sub>i,j max</sub>
0	15	1	5	5
1	13	4	2	2
3	10	3	4	4
4	8	4	3	4
5	7	1	0	4
7	4	3	0	6
8	4	4	0	5

Tabla 5.3: Resultado de eliminar el canal 2.

Este esquema sirve a la vez para asignar los valores  $C_R$  y mantener los intervalos en el nodo. El coste de insertar y eliminar un canal será por tanto  $O(m*\log(nm))$ . Este esquema también serviría en el caso de que el valor  $C_R$  variase. Simplemente, habría que subir o bajar el listón hasta que el valor  $C_R$  fuese 0.

### 5.6.- Resultados comparativos y pruebas

El objetivo de este punto es comparar la reserva total de *buffer*  $B$  realizada por el nuevo método introducido (a partir de aquí se denominará *Puntos*) comparándolo con el de Presti (método *Greedy*) utilizando tráfico MPEG. Ya que se desconoce las cargas utilizadas en el artículo de Presti et al. se utilizarán cinco tipos de cargas MPEG (LAMBS, NEWS, SOCCER, ASTERIX y STARS). Para esta comparación se han implementado en el simulador RTNetSim (véase el apéndice 2) los dos algoritmos de optimización del *buffer*.

Las pruebas se han planteado de la siguiente forma:

1. Para cada clase de tráfico se definen cuántos canales va a tener. Esto permite aumentar la carga de la red fácilmente sin utilizar muchos tipos de tráfico. Por ejemplo, en la primera comparación se utilizan sólo dos tráfico: LAMBS y SOCCER con 10 canales cada uno, lo que da una carga total de 20 canales.
2. Después se obtiene para un retraso determinado  $d$ , los valores  $\sigma$  y  $r$  que optimizan la reserva de ancho de banda en la red para cada clase de tráfico. Para ello se define una red de prueba con todos los nodos iguales, con enlaces de 100Mbps. Los valores  $MTU$  y  $M$  de los nodos se han fijado a 10000 bits.
3. Con los valores de  $\sigma, r$  y la tasa pico  $p$  se calcula el valor  $T_{on}$  para cada clase.
4. Con todos estos valores se confecciona una tabla que contiene los datos más significativos de la prueba: número total de cargas, valores  $\sigma$  y  $r$  calculados, tasa pico, valor  $T_{on}$  y número de puntos que se necesitan enviar al nodo.

5. Se varía el valor de  $C$  entre el mínimo ancho de banda requerido hasta al máximo (la suma de todas las tasas pico), calculando el valor de  $B$  en cada esquema

El cálculo de los valores  $\sigma, r$  a partir del retraso no es estrictamente necesario ya que se podrían usar otros valores y con estos hacer las pruebas. Se han utilizado por ser valores óptimos y fáciles de calcular. Además, esto permite comparar las reservas en función del retraso máximo. La red usada es de dos nodos, pero ésta se usa sólo con el objetivo de calcular los parámetros  $C_{tot}$  y  $D_{tot}$  para poder obtener el retraso de la red. Por ello, el número de nodos de la red y sus características no influyen en los resultados.

A efectos comparativos también se ha simulado la reserva media y máxima de memoria en los nodos. Ya se ha comentado que la reserva real dependerá de las trazas y de la distribución temporal en ellas. Para poder calcular valores medios y máximos se ha repetido el cálculo de la memoria usada variando el comienzo de las distintas trazas. Así, dado un valor  $C$ , el cálculo del valor medio y máximo se realiza de la siguiente forma:

1. Se obtienen el valor  $r$  para cada canal utilizando el método de reserva óptima.
2. Con los valores de  $r$  se obtiene la memoria utilizada usando las ecuaciones (1) y (2), y las trazas de las cargas.
3. Se repite el segundo paso 20 veces variando el comienzo de las trazas.

Para que las tramas varíen en el tiempo se asigna el siguiente valor de comienzo de trama, es decir, cuántas tramas tiene que esperar el canal  $i$  para empezar a transmitir.

$$Comienzo_i = i * n^{\circ} Iteracion * 5$$

De esta forma, las trazas se van desplazando entre sí de 5 en 5 en la primera iteración, y este desplazamiento se va incrementando en cada iteración, con lo cual se obtiene una variación sin repeticiones.

Hay que recalcar que estos valores son obtenidos simplemente para permitir su comparación con los esquemas analizados (*Puntos* y *Greedy*). Si se simulasen todas las combinaciones temporales de los tráfico, el valor máximo sería igual al obtenido por el método desarrollado. Esta combinación es difícil que ocurra: si se tienen 20 tráfico de 40000 tramas la probabilidad de que todos los picos coincidan sería de  $1/40000^{20} = 9 * 10^{-93}$ .

A continuación, se detallan los resultados de cuatro pruebas. En las dos primeras se utilizan dos clases de cargas y en las segundas cinco.

### 5.6.1.- Resultados con dos clases

En la primera comparación se usan dos clases de cargas, LAMBS y SOCCER, con diez canales cada una. Se calculan los valores de  $\sigma, r$  para un retraso de 0.1s que se muestran en la tabla 5.4. Esto nos da un rango de valores de  $C$  de [35112550, 80350000]. El primer valor es la suma de las reservas mínimas (es decir  $r$ ) de todos los canales. El segundo valor es la suma de las tasas pico de todos los canales. En la figura 5.6 se ve el *buffer B* necesario en función de ancho de banda  $C$  del nodo. Como se muestra, el ahorro del esquema *Puntos* sobre el *Greedy* es importante para valores de  $C$  entre  $3.5 \cdot 10^7$  y  $5.5 \cdot 10^7$  para coincidir luego a partir de  $5.7 \cdot 10^7$ . Además, se puede comprobar que el número de puntos a enviar es mínimo (última columna de la tabla 5.4). El número de puntos define también el número de intervalos a la hora de optimizar.

Con respecto a la memoria pico y media de las simulaciones, se muestra claramente que la memoria pico tiene un patrón parecido a la curva de la optimización por puntos y que no hay mucha diferencia entre ambas. Respecto a la memoria media usada, el mayor valor lo da al principio con mucha diferencia con respecto a la curva de optimización por puntos y luego va decayendo muy lentamente.

Carga	N	$\sigma$ (bits)	$r$ (b/s)	Tasa pico(bits)	$T_{on}$ (s)	Nº puntos
Lambs	10	90096	1103175	3355600	0.0400	1
Soccer	10	220326	2408080	4679400	0.0968	4
<b>Total</b>	<b>20</b>	<b>3104220</b>	<b>35112550</b>	<b>80350000</b>		<b>5</b>

Tabla 5.4: Valores de  $\sigma, r$  para un retraso de 0.1s.

En una segunda comparación, se aumenta el retraso a 1s con lo que los valores de  $\sigma$  aumentan considerablemente (tabla 5.5). Esto da un rango de valores de  $C$  entre [18802550, 80350000]. En este caso la ganancia es bastante mayor y las dos curvas (*Puntos* y *Greedy*) no coinciden en ningún punto (véase la figura 5.7). Respecto a la prueba anterior se comprueba que la diferencia entre *Puntos* y *Greedy* ha aumentado considerablemente.

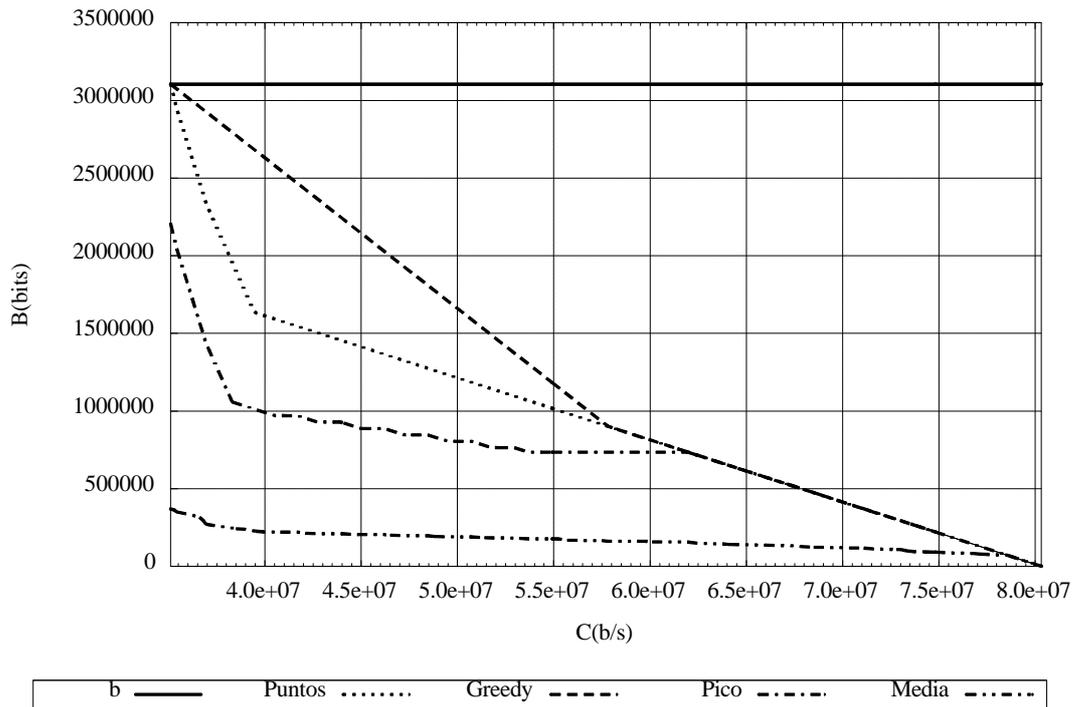


Figura 5.6: 2 clases y  $d = 0.1s$ .

Carga	N	$\sigma$ (bits)	$r$ (b/s)	Tasa pico(bits)	$T_{on}$ (s)	Nº puntos
Lambs	10	615540	635667	3355600	0.2263	10
Soccer	10	1226138	1246388	4679400	0.3572	7
<b>Total</b>	<b>20</b>	<b>18416780</b>	<b>18820550</b>	<b>80350000</b>		<b>17</b>

Tabla 5.5: Valores de  $s, \rho$  para un retraso de  $1s$ .

En la figura 5.8 se muestra el ahorro de recursos para distintos valores de retraso y ancho de banda. Cuanto mayor es el retraso, mayor es la ganancia, obteniendo valores cercanos al 95%. Esto es así debido a que cuanto mayor es el retraso los valores de  $T_{on}$  son mayores (debido a que el cociente  $\sigma/r$  se incrementa), y esto hace que la aproximación del modelo *Greedy* al tráfico sea más burda. Se comprueba que para retrasos altos el incremento de ganancia aumenta rápidamente y se mantiene constante en valores alrededor del 90%. Para retrasos más pequeños hay un pico de ganancia que varía entre  $3.7 \cdot 10^7$  a  $4.7 \cdot 10^7$ . Las pequeñas ganancias en retrasos pequeños son debidas principalmente a que la reserva es elevada y el número de puntos es muy reducido o sólo hay uno, como en el ejemplo de la tabla 5.4.

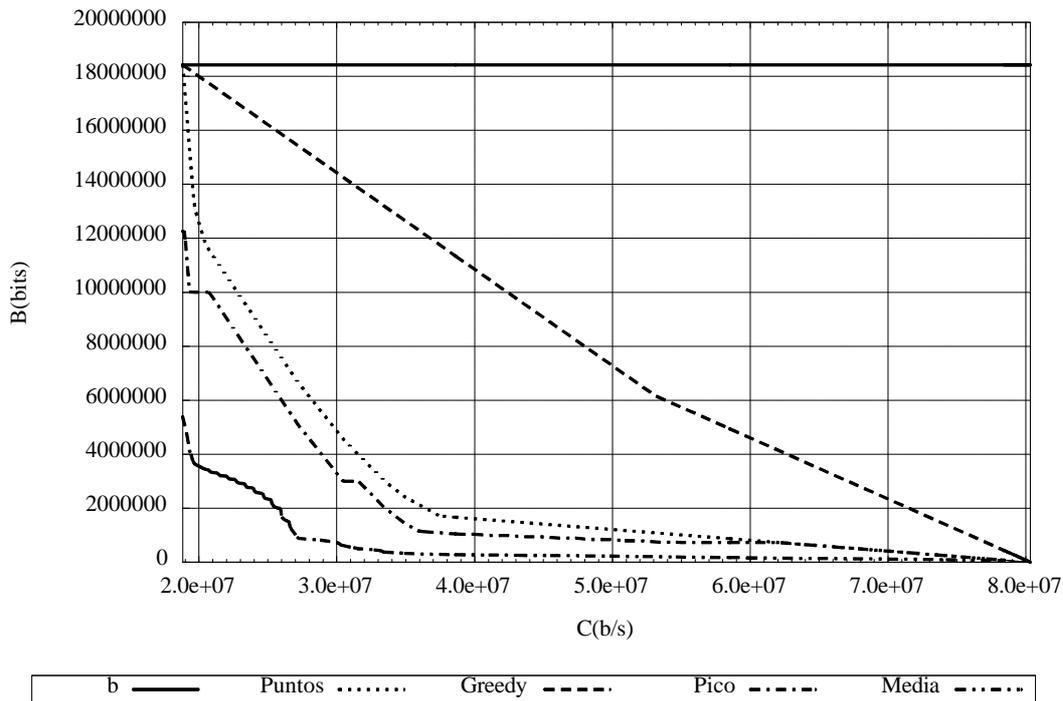


Figura 5.7: 2 clases y  $d = 1s$ .

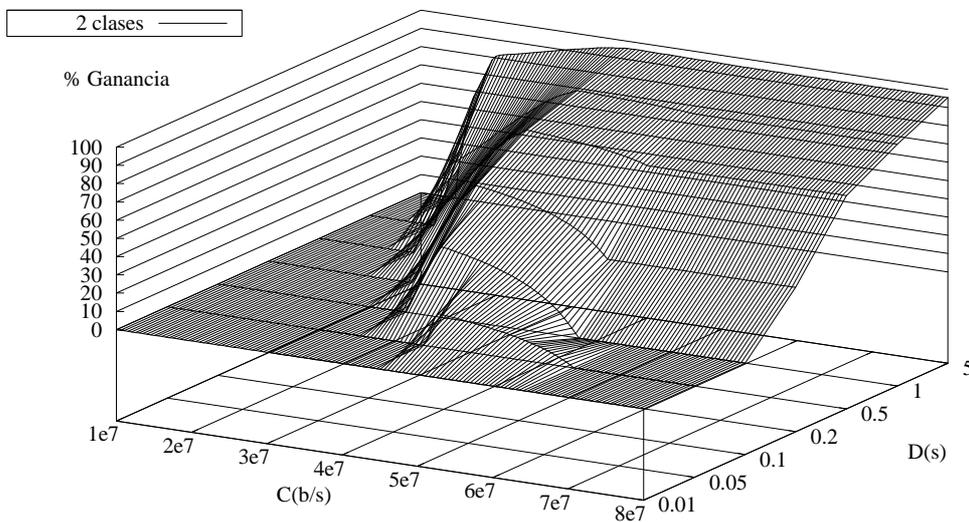


Figura 5.8: Ahorro de Buffer en función de C y retraso para 2 canales.

### 5.6.2.- Resultados con cinco cargas

En las siguientes pruebas se utilizan cinco clases de cargas, con cinco canales cada una. En la figura 5.9 se calcula  $\sigma, r$  para un retraso de 0.1s (tabla 5.6), siendo el rango de

valores de  $C$  [35952620, 79974000]. En la figura 5.10 se muestran los resultados para un retraso de 0.5s con lo que los valores de  $\sigma$  se incrementan considerablemente (tabla 5.7) y el rango de  $C$  es [24033130 79974000]. Los resultados son muy parecidos al grupo de dos clases. El hecho más significativo es que la curva de puntos no se muestra tan segmentada como en el caso de 2 clases, lo cual es lógico al tener muchos más intervalos que seleccionar.

Carga	N	$\sigma$ (bits)	$r$ (b/s)	Tasa pìco(bits)	$T_{on}$ (s)	Nº puntos
Lambs	5	90096	1103175	3355600	0.0400	1
Soccer	5	220326	2408080	4679400	0.0970	4
News	5	40288	604093	1155000	0.0731	2
Asterix	5	183521	2039297	3684400	0.1116	5
Star2	5	83380	1035879	3120400	0.0400	1
<b>Total</b>	<b>25</b>	<b>3088055</b>	<b>35952620</b>	<b>79974000</b>		<b>13</b>

Tabla 5.6: Valores de  $s, p$  para un retraso de 0.1s.

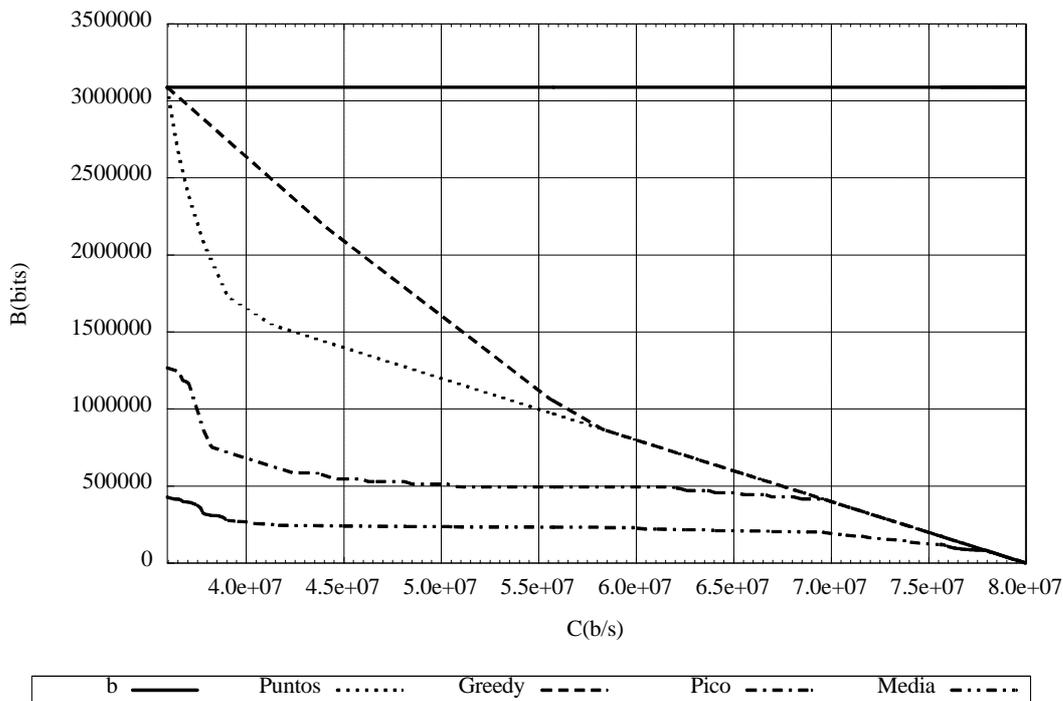


Figura 5.9: 5 clases y  $d = 0.1s$ .

Con respecto a las simulaciones, en ambos casos se puede ver que existe mayor diferencia entre optimización por puntos y el *buffer* máximo que en las pruebas con dos cargas. Esto es debido al aumento del número de clases (se pasa de 2 a 5) lo cual provoca que la tasa pico esté más cercana a la media. Además, en el caso de dos trazas, en la primera iteración todas las trazas empiezan al mismo tiempo lo que provoca que todos los picos de cada clase de tráfico coincidan. En este caso, aunque también coinciden en la primera iteración, al ser cinco clases el acumulado no está tan cercano al máximo posible.

Carga	N	$\sigma$ (bits)	$r$ (b/s)	Tasa pico(bits)	$T_{on}$ (s)	Nº puntos
Lambs	5	321722	683719	3355600	0.1204	8
Soccer	5	771405	1583444	4679400	0.2492	6
News	5	89696	219479	1155000	0.0959	3
Asterix	5	744541	1529695	3684400	0.3455	10
Star2	5	374986	790289	3120400	0.1609	9
<b>Total</b>	<b>25</b>	<b>11511750</b>	<b>24033130</b>	<b>79974000</b>		<b>36</b>

Tabla 5.7: Valores de  $s, \rho$  para un retraso de 0.5s.

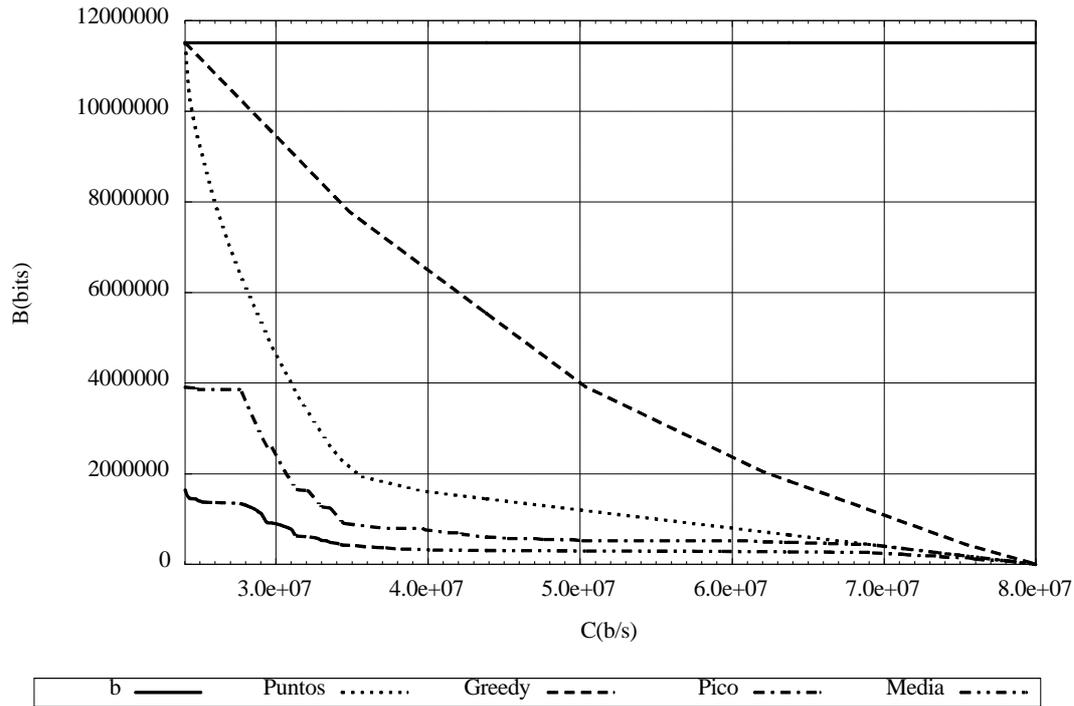


Figura 5.10: 5 clases y  $d = 0.5s$ .

En la figura 5.11 se ve el ahorro de *buffer* obtenido con respecto al esquema de Presti para varios retrasos. Esta gráfico es bastante similar a la de dos canales.

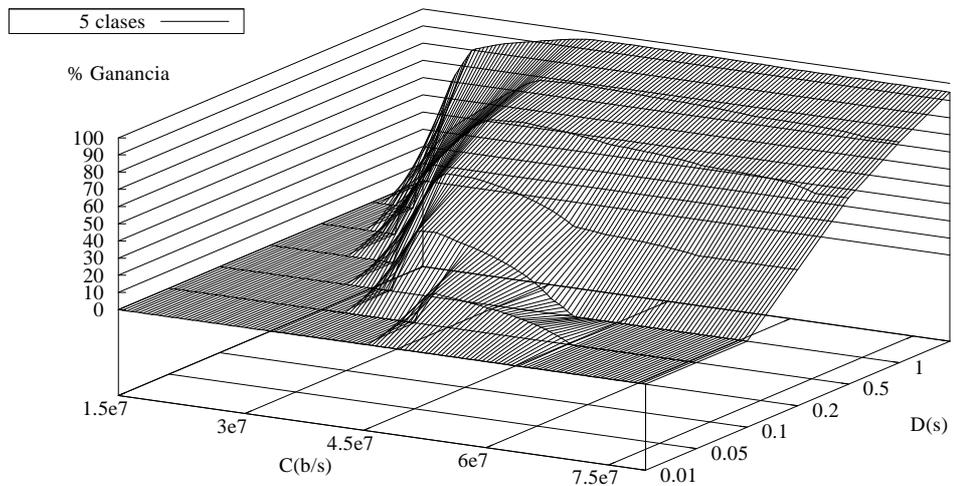


Figura 5.11: Ahorro de Buffer en función de  $C$  y retraso para 5 canales.

### 5.6.3.- Conclusiones

Como muestran las pruebas realizadas, el esquema introducido mejora bastante el presentado por Presti. En general, esta mejora es mayor cuando se trabaja con retrasos altos, debido a que la reserva es menor y se envían más puntos, y el modelo de tráfico *Greedy* se aproxima menos al tráfico original.

Con respecto a las simulaciones, se muestra que la diferencia con el *buffer* medio usado se incrementa cuando el número de canales distintos aumenta, lo cual es lógico, ya que al aumentar el número de muestras (canales), la diferencia entre el máximo y la media aumenta.

### 5.7.- Aplicación a varios nodos

En los puntos anteriores se ha visto cómo se optimiza la memoria en un nodo con el tráfico regulado a la entrada. Este planteamiento es el mismo que el utilizado por Elwalid et al. y Presti et al. en sus artículos. Pero tal como se comenta en el artículo de Elwalid, este esquema encuentra dificultad de aplicación a un entorno de red con varios nodos. En este artículo se comenta que, si bien la tasa pico puede aumentar en teoría, las pruebas realizadas por ellos siempre han producido un suavizado del tráfico y que por tanto el cambio de las características del tráfico al pasar por los nodos es insignificante, con lo que no lo tienen en cuenta. Resumiendo, ninguno de los dos artículos comentados aportan soluciones a la aplicación a varios nodos.

El problema es que, mientras sí que es posible caracterizar el tráfico a la entrada de una red, este tráfico puede ser distorsionado dentro de la red, con lo que la caracterización realizada de la fuente no puede ser aplicada en los nodos internos a la red [Zhang98]. En general, existen tres soluciones para gestionar el problema de la distorsión del tráfico:

1. Controlar la distorsión del tráfico dentro de la red. Esto implica poner conformadores de tráfico (*traffic shapers*) en determinados puntos en la red. Esto supone usar disciplinas de servicio *non-work-conserving*.
2. Gestionar esta distorsión en la red. Se puede indicar al planificador que trabaje con tiempos lógicos de llegada de paquetes y retrasar los paquetes que lleguen antes del tiempo lógico.
3. Caracterizar la distorsión del tráfico en la red. El problema se puede formular de la siguiente forma: dada una caracterización del tráfico de todas las conexiones en la entrada a la red y de todas las disciplinas de servicio en los nodos, ¿puede ser caracterizado el tráfico para cada conexión en cada nodo de la red?.

De las soluciones planteadas en esta tesis se han abordado la primera y la tercera con éxito. La segunda no se ha planteado por que implicaría utilizar otro tipo de planificadores.

### **5.7.1.- Conformación del tráfico en la red**

En la figura 5.12 se puede ver un ejemplo ficticio de cómo varía el tráfico cuando pasa por distintas redes. Esta red contiene una red local 802.x y dos redes públicas ATM. En la parte superior de esta figura se representa la variación del tráfico al pasar por las distintas redes como gráficas de ancho de banda en función del tiempo. En el gráfico (a) se tiene el tráfico original. Este tráfico es distorsionado al pasar por la red local y llegar al *router* (b). Al pasar por el conformador vuelve a tener las mismas características que el tráfico origen (c). Al pasar por las redes públicas 1 y 2 se distorsiona de nuevo (d)(e). Cuando el tráfico llega al receptor, el tráfico se puede conformar para recuperar las características del tráfico original (f).

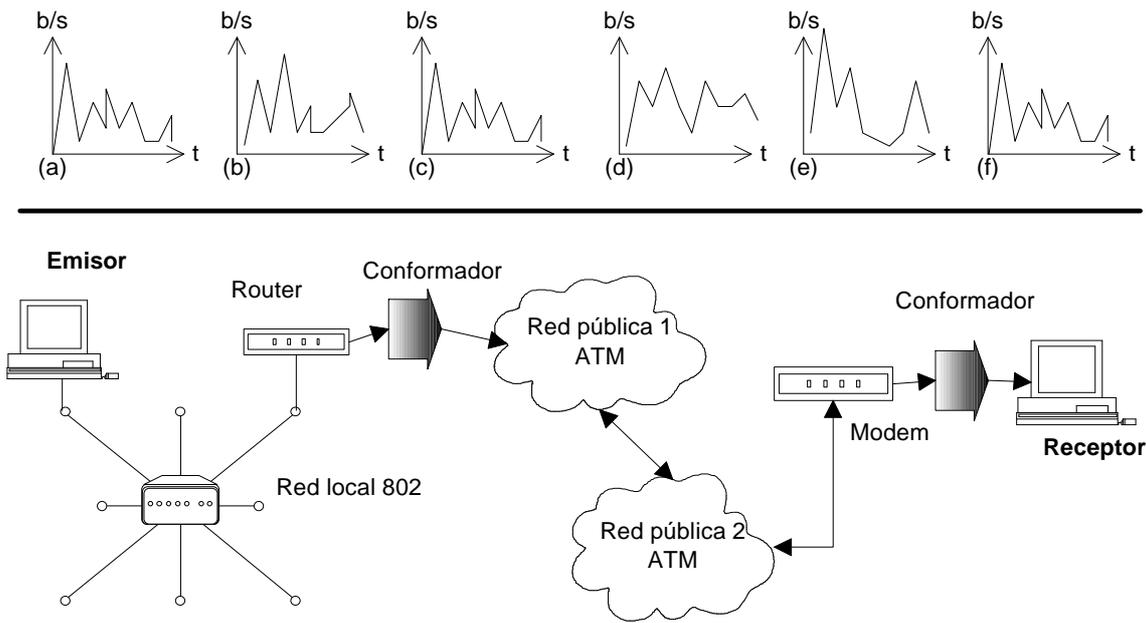


Figura 5.12: Variación del tráfico en una red.

Intentar que el conformador de tráfico recupere exactamente el tráfico origen es una tarea prácticamente inviable. Pero esto no es necesario, ya que lo único que se requiere es que este tráfico conformado tenga los mismos puntos envolventes para entonces poder aplicar el esquema de optimización diseñado. Esto se puede hacer con un conformador de tráfico *leaky bucket* multinivel como se describe a continuación.

Al utilizar los puntos envolventes se está definiendo una función de tráfico limitado que está formada por los segmentos que componen la función  $s'(r)$ :

$$l(t) = \min_{0 \leq i < m} \{s_i + r_i t\} \quad \forall t \tag{9}$$

Cualquier tráfico que sea conforme a esta función de tráfico limitado tendrá los puntos envolventes del tráfico original como se puede ver en la figura 5.13.

Para conformar un tráfico de esta manera se puede usar un conformador *leaky bucket* multinivel, tal como se comenta en [Knightly95]. La implementación de este conformador se realiza con  $n$  conformadores *leaky bucket* en cascada como prueba R. Cruz en el teorema 5.1 [Cruz91b].

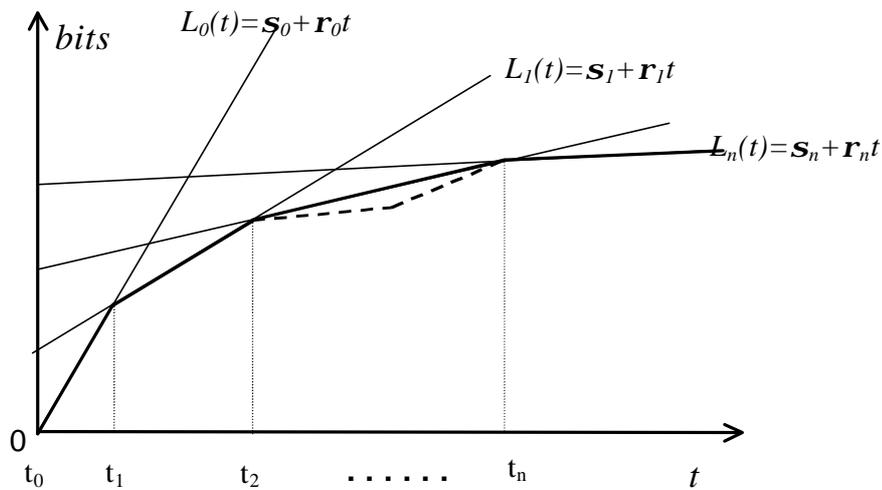


Figura 5.13: Función de tráfico limitado.

La introducción de un conformador en la red no implica una variación en el retraso máximo emisor-receptor de un paquete tal como demuestra Zhang [Zhang93]. Esto es lógico, ya que lo único que realiza el conformador es retrasar los paquetes que no cumplen la ecuación 8, y si no lo cumplen es porque la diferencia de tiempo entre paquetes se ha reducido, lo que implica que han llegado antes de tiempo al conformador.

Por lo tanto, poniendo un conformador de tráfico en los nodos, podemos aplicar el esquema de optimización visto. El problema reside en que el conformador necesita memoria para almacenar los paquetes que no están conformes, con lo que el ahorro de memoria en los nodos se tendría que utilizar en el conformador. La memoria necesaria en este conformador es la misma que la utilizada en el planificador con la tasa de drenaje original del tráfico, lo que en conjunto, no supone ningún ahorro en el nodo (a no ser que por diseño esto implique una mejora: por ejemplo, con el objetivo de conseguir altas velocidades de transmisión, el planificador de paquetes podría ser un componente hardware que tendría los recursos más limitados y la conformación de los tráficos hacerse antes de la entrada del planificador en un ordenador con gran capacidad de memoria. El ejemplo típico de este esquema sería un nodo ATM de alta velocidad).

Por lo tanto, para que este esquema sea eficiente, es importante no utilizar demasiados conformadores. Para ello, se pueden introducir los conformadores a la entrada de subredes y optimizar en estas subredes balanceando el uso de memoria y ancho de banda. Así, siguiendo el ejemplo de la figura 5.12, se puede optimizar a la entrada de la red pública 1. Para lograrlo se agruparán varios canales con un mismo origen-destino dentro de la subred en un solo macrocanal. A este macrocanal se le asignará un ancho de banda  $C$  para toda la subred y el objetivo será por tanto optimizar la memoria  $B$  necesaria para ese

macrocanal. Este problema es equivalente al de un nodo con lo que se aplicará el mismo esquema y algoritmos vistos en el punto 5.4. De esta manera, el ahorro se producirá en toda la subred a costa de introducir un único conformador a la entrada de la subred.

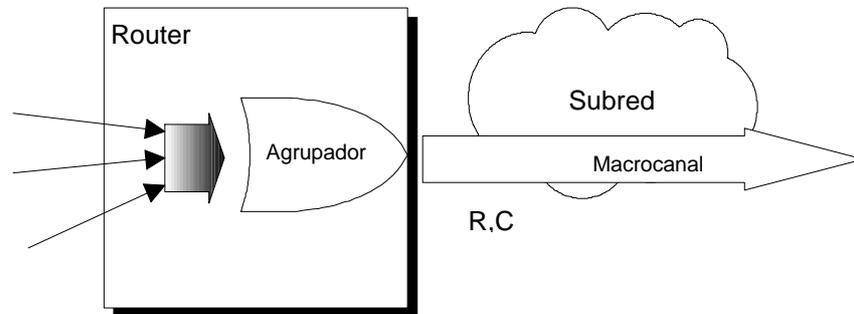


Figura 5.14: Macrocanal.

La asignación del ancho de banda  $C$  podría ser dinámica y negociada con la subred. En consecuencia, si la red estuviera poco congestionada podría proporcionar más ancho de banda con lo que las necesidades de memoria del macrocanal disminuirían. Si la red estuviera muy congestionada, el ancho de banda  $C$  bajaría hasta el mínimo necesario para asegurar el retraso negociado en la reserva de recursos (la suma de la reserva  $R$  de todos los canales). Esta gestión dinámica no sería costosa de implementar, ya que utilizando el algoritmo visto en el punto anterior, un aumento o disminución de  $R$  sólo implicaría subir o bajar el listón en la tabla de canales hasta que el ancho de banda disponible sea cero.

La agrupación de todos estos tráfico en un macrocanal no implica la planificación conjunta de todos los canales. Cada canal será planificado por separado asignándole la nueva reserva  $c_i$  calculada usando el algoritmo 5.5. Lo que sí que permite esta agrupación es saber que el conjunto de estos canales va a utilizar una memoria determinada  $B$  menor que la que en principio tenían asignada.

### 5.7.2.- Caracterización de la distorsión en la red.

Tal como se comenta en [Zhang98], caracterizar el tráfico dentro de la red es difícil y puede no ser siempre posible. En [Cruz91b][Parekh93] se muestra que la solución consiste en resolver un conjunto de ecuaciones con múltiples variables pero hay muchas situaciones en la que este sistema de ecuaciones no tiene solución.

En este punto se demostrará que es posible utilizar en los nodos intermedios de la red los puntos envolventes para optimizar la memoria. La solución óptima sería caracterizar el tráfico a la entrada de un nodo y poder calcular los puntos envolventes. Esto

actualmente es inviable como se comenta en el párrafo anterior. Por lo tanto, lo que se obtiene es una solución sub-óptima.

Para ello, al igual que se comenta en el punto anterior 5.7.1, no es necesario caracterizar perfectamente el tráfico dentro de la red, sino obtener una cota superior de estos puntos envolventes al pasar por un nodo. Esto permitiría aplicar las optimizaciones vistas en cualquier nodo de la red. La solución se basa en las propiedades de las ecuaciones de retraso.

Como se ha visto en el segundo capítulo, las ecuaciones de retraso de IETF permiten obtener el *buffer* necesario en un nodo  $j$ . El tamaño del *buffer* en un nodo vendrá determinado por la siguiente ecuación:  $\mathbf{s} + C_{sum} + D_{sum}\mathbf{r}$  donde  $C_{sum}$  y  $D_{sum}$  son el acumulado de los términos de error  $C$  y  $D$  de los nodos anteriores. Lo que se produce es un incremento al tamaño del cubo  $\mathbf{s}$  que depende del orden del nodo en la ruta establecido. Este incremento se representará como  $\mathbf{d}$ .

Por lo tanto, el *buffer* necesario en el nodo  $j$  se puede expresar como  $\mathbf{s} + \mathbf{d}^j$ . Hablando en términos de modelo de tráfico *token bucket*, esta ecuación lo único que refleja es que, el tráfico puede aumentar el tamaño de una ráfaga (el valor  $\mathbf{s}$ ) con lo que es necesario aumentar la memoria para almacenar el tráfico. Por lo tanto, el tráfico a la entrada del nodo  $j$  estará limitado por los nuevos valores  $(\mathbf{s} + \mathbf{d}^j, \mathbf{r})$ , es decir, la función de tráfico limitado será  $(\mathbf{s} + \mathbf{d}^j) + \mathbf{r}t$  [Cruz95].

Esta propiedad se puede aplicar a los puntos envolventes. Como se ha comentado en el punto anterior, al usar los puntos envolventes se está definiendo una función de tráfico limitado (ecuación 9 y figura 5.13). Como se puede ver, existen  $n$  funciones que se corresponden a  $n$  *token bucket*. Aplicando la propiedad vista en el párrafo anterior, la nueva función de tráfico limitado para el nodo  $j$  es la siguiente:

$$l^j(t) = \min_{0 \leq i < m} \{(\mathbf{s}_i + \mathbf{d}_i^j) + \mathbf{r}_i t\} \quad \forall t \quad (10)$$

Tal como se ve en la figura 5.15 esto supone un desplazamiento de la función hacia arriba. Esto implica que la posición de los puntos envolventes se mantiene, lo único que varía es el valor  $E(t_i)$ . Como se ha visto, este valor no es necesario para los algoritmos descritos ya que lo realmente necesario es la pendiente de las rectas y la posición de los puntos, y éstos no varían.

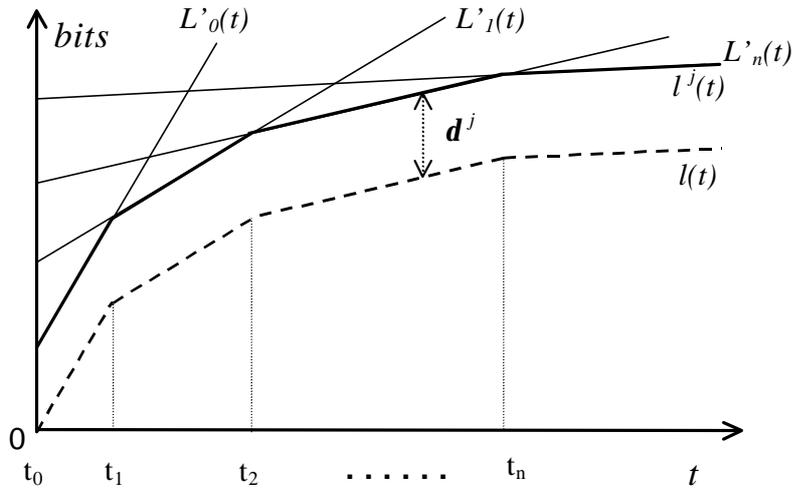


Figura 5.15: Variación de la función de tráfico limitado.

Lo que está ocurriendo es que los puntos envolventes se están desplazando hacía arriba con lo que aumenta el valor del *buffer* calculado. Hay que indicar que se está hablando siempre de cotas máximas y que en este caso se trata de una solución sub-óptima. Lo único que se asegura es que el tráfico siempre está dentro de la función de tráfico limitado asociada a los puntos envolventes. Pero claro, esto no es la mejor aproximación al tráfico que entra en un nodo. Para ello, se tendría que obtener los puntos envolventes del tráfico a la entrada del nodo lo cual es prácticamente inviable.

Resumiendo, para calcular la memoria necesaria de cada canal se tendría que sumar a los valores  $c_j$  el valor  $d^j$ . Así, utilizando los ejemplos de la tabla 5.7 con una red en la que el ancho de banda es 100Mbps, y el *MTU* y *M* es de 10000 bits, se tiene que los valores *C* y *D* en cada nodo son:

$$D = \sum \frac{MTU}{AB} = \frac{10000}{100 * 10^6} = 0.0001 \quad C = M = 10000$$

y dado que el tamaño del *buffer* se incrementa en  $C_{sum} + D_{sum}r$ , el valor  $d^j$  es:

$$d^j = (C + Dr)(j-1)$$

con lo que para el tráfico LAMBS con un valor de *r* igual a 683719b/s el *buffer* necesario se incrementará en 10069 bits cada nodo. Así, en el quinto nodo la memoria calculada se incrementará en 40276 bits. Para el tráfico SOCCER del ejemplo de la tabla 5.6 que tiene un valor de *r* igual a 2408080 b/s el incremento en cada nodo será 10240. Como se ve en estos ejemplos el valor que más afecta al incremento de la memoria es el tamaño máximo del paquete de un canal.

### 5.8.- Balanceo en la optimización

Utilizando los puntos envolventes se puede, primero optimizar la reserva de ancho de banda para un determinado canal, y a continuación optimizar la memoria (*buffer*) de los nodos teniendo en cuenta la información de los canales. De esta forma, se puede balancear el consumo de memoria en los nodos con el ancho de banda reservado. En este punto se muestra un ejemplo de cómo realizar este balanceo.

En la figura 5.16 se muestra una red compuesta por tres emisores y dos receptores. Todos los tráficos entre estos emisores y receptores pasan por una subred que tiene un conformador de tráfico a su entrada. Con el objeto de poder balancear la memoria y ancho de banda en la subred, se crea un macrocanal con todo el tráfico entre estos emisores y receptores. Esta subred, que puede representar una red pública, está compartida con otros tipos de tráficos. Los parámetros de la red están en la tabla 5.8.

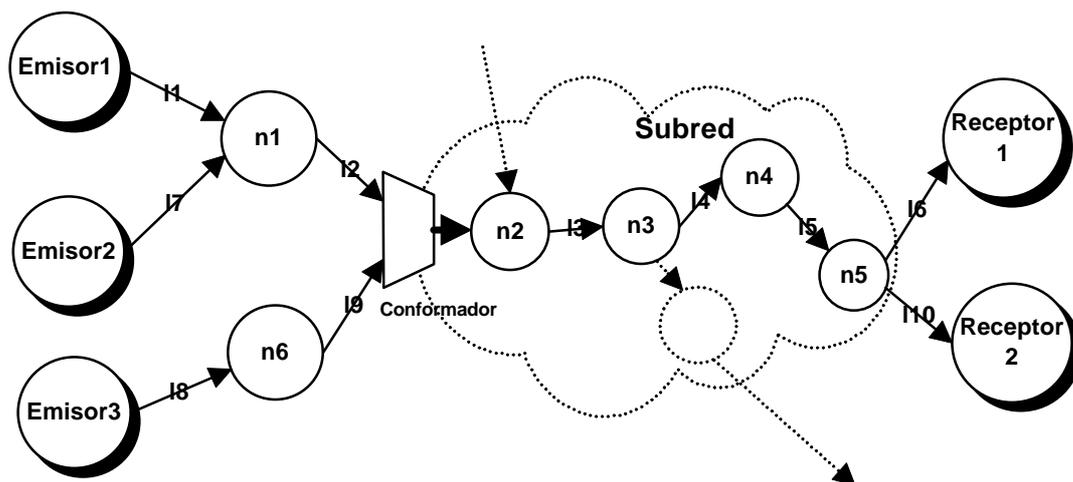


Figura 5.16: Red ejemplo con una subred.

Enlace	Ancho de Banda (Mbps)	MTU(bits)
11,17,18	10	10000
12,19	15	15000
13	50	10000
14	100	12000
15	55	10000
16,110	15	10000

Tabla 5.8: Parámetros de la red de la figura 5.16.

En la red de la figura 5.16 se van a transmitir los tráficos indicados en la tabla 5.9 con el origen y el destino indicados.

Tráfico	Origen	Destino
Lambs	Emisor1	Receptor1
Soccer	Emisor1	Receptor1
News	Emisor2	Receptor1
Asterix	Emisor3	Receptor2
Star2	Emisor3	Receptor2

Tabla 5.9: Origen y destino de los tráfico.

Lo primero que se hace, es calcular la reserva mínima a realizar para cada tráfico en la red dado un retraso máximo de 0.01s. Los resultados están en la tabla 5.10. En total, se necesita un ancho de banda  $C$  mínimo de 8196865b/s para asegurar el retraso con un tamaño total de *buffer*  $B$  total de 493701bits.

Tráfico	$\sigma$ (bits)	$r$ (b/s)	Tasa pico(bits)	Nº puntos
Lambs	77445	1419439	3355600	1
Soccer	181706	2496193	4679400	3
News	21380	840445	1155000	2
Asterix	142382	2090076	3684400	4
Star2	70788	1350697	3120400	1
<b>Total</b>	<b>493701</b>	<b>8196865</b>	<b>15994800</b>	<b>11</b>

Tabla 5.10: Reserva mínima para un retraso máximo de 0.1s.

Si a la entrada de la subred se agrupan todos estos canales en un macrocanal y se le asignan 10 Mb/s a  $C$  esto supone que el valor  $B$  se reduce a 239791bits. La reducción del valor  $B$  en función del valor  $C$  asignado al macrocanal podrá variar entre la reserva mínima necesaria para todos los canales: 8196865b/s hasta la suma de las tasas pico: 15994800b/s que está representado en la figura 5.17.

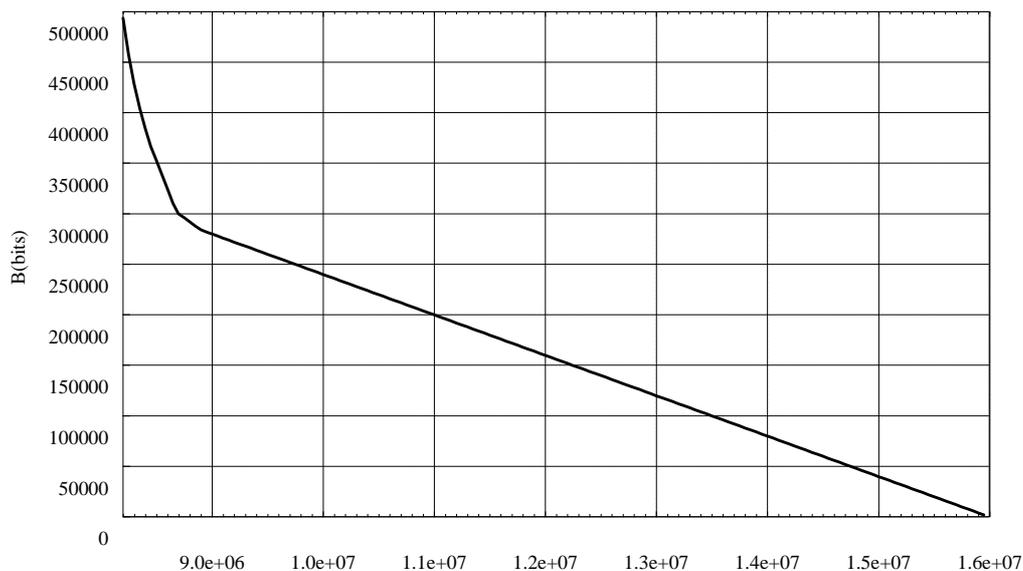


Figura 5.17: Memoria en función de la reserva para la subred.

Esto permite modificar las necesidades de memoria en los nodos de la subred de forma dinámica. Por ejemplo, si se dispone de un ancho de banda de 1200000b/s para el macrocanal esto implicará un uso de memoria de 1700000bits tal como se puede ver en la figura 5.17.

## 5.9.- Conclusiones

En este capítulo se ha mostrado un nuevo esquema para optimizar la reserva de memoria en los nodos basado en los puntos envolventes. Este método es mucho más eficiente que el basado en el modelo de tráfico *Greedy* propuesto por Presti et al.

Este nuevo esquema se basa en poder enviar los puntos envolventes a los nodos con el objeto de poder optimizar de forma conjunta la memoria utilizada por los canales. Esta solución asume, de igual forma a las propuestas por Elwalid y Presti, que el tráfico está conformado a la entrada del nodo.

Además, se proponen dos soluciones para su aplicación en varios nodos:

1. La primera está basada en conformar el tráfico a la entrada de un nodo para recuperar las características del tráfico original en lo que respecta a la posición de los puntos envolventes. Para ello, se utilizará un conformador *leaky bucket* multinivel. Además, se propone agrupar los canales en un macrocanal, lo que permite balancear entre la optimización de la reserva de ancho de banda y la memoria necesaria.
2. En la segunda, se establece una cota a la distorsión del tráfico, lo que implica un conjunto de puntos envolventes que a efectos de optimizar es equivalente al original. Esta solución es sub-óptima ya que la óptima sería tener los puntos envolventes del tráfico a la entrada de los nodos, lo que es inviable.

Durante el capítulo se han expuesto varios ejemplos con tráfico MPEG que muestran que la mejora en la reserva de memoria es bastante amplia con respecto al esquema basado en un modelo del tráfico *Greedy* (puede llegar hasta el 95%).

Dado que los puntos que se envían a los nodos son muy pocos, este esquema no es complejo de implementar ni sobrecargaría los nodos.

## 6.

# Reserva eficiente de recursos para canales de reserva

### 6.1.- Introducción

Como se ha visto en capítulos anteriores, la calidad de servicio se establece entre el cliente y el proveedor del servicio para garantizar determinados parámetros como rendimiento, retraso máximo o fiabilidad. Sin embargo algunas aplicaciones de tiempo real críticas, control de procesos, monitorización científica y médica, etc. necesitan además disponibilidad: mantener la calidad de servicio a pesar de fallos en los nodos o enlaces.

El esquema de reserva de recursos y control de admisión ha tenido éxito al proporcionar tiempos de transmisión garantizados, pero es difícil evitar los errores debidos a fallos en los nodos. Esto es debido a que los mensajes sólo pueden ir por la ruta reservada a priori, y no pueden sortear los elementos con fallos de forma dinámica. Esto se ha solucionado utilizando canales redundantes con dos esquemas opuestos [Han97]: en la copia múltiple (*Multiple Copy* o *Forward Recovery Approach*) [Ramanathan92][Kao94] todos los canales redundantes están activos simultáneamente y los mensajes son enviados a la vez por todos los canales para enmascarar los efectos de los fallos en la red. El otro esquema es establecer un canal de reserva (*Backup Channels Approach*) [Banerjee93] que sólo se activa cuando se detecta un fallo. Este segundo esquema parece más atractivo, ya que no gasta recursos en la ausencia de fallo, pero para que sea efectivo el canal secundario se tiene que establecer de forma rápida y segura. Esto implica que también se tengan que reservar los recursos en la red aunque no se usen.

Para reducir esta reserva de recursos es muy interesante el esquema introducido por Zhang y Shin basado en reservar los recursos sobrantes (*spare resources*) de los elementos vecinos del componente que ha fallado [Zheng92]. Esquemas similares han sido utilizados para servicios de telecomunicaciones fiables [Grover87][Baker91]. El problema de estos

mecanismos de enrutamiento local es que pueden alargar la ruta entre emisor y receptor con lo que se desperdician recursos. El enrutamiento emisor-receptor soluciona este problema y además requiere menos recursos [Han97]. Para ello, se reserva un canal de reserva con los recursos sobrantes de los nodos. Esto implica que estos recursos estén reservados pero no son utilizados hasta que se active el canal de reserva. Esto permite una activación rápida del canal de reserva, que no será tan rápida como si estuvieran realmente reservados.

Sin embargo, existe un problema por resolver: los requerimientos de retraso en el canal de reserva tienden a ser muy exigentes, debido a la latencia en la detección del fallo y el tiempo de establecimiento del canal.

Se ha visto que la reserva de recursos se puede obtener a partir de una caracterización de la carga y la red dado un retraso determinado. Además del retraso exigido, un componente determinante es el esquema de detección del fallo. Esto es debido a la siguiente observación: cuanto mayor es el tiempo de latencia de fallo, mayor es el requerimiento de retraso exigido al canal de reserva para poder mantener la calidad de servicio. Por ello, en este capítulo se introduce un nuevo esquema para la detección del fallo cuyo objetivo es reducir la latencia del fallo (y por lo tanto, reducir la reserva). La técnica propuesta reduce la latencia “sospechando” un fallo antes de que ocurra. Esto provoca activaciones innecesarias del canal de *backup* cuando un fallo al final no es confirmado. Esta técnica puede ser considerada un compromiso entre los dos esquemas descritos. La demanda de recursos de esta técnica se compara con los esquemas previos usando el protocolo RSVP usando tráfico MPEG. Se evalúa el porcentaje de activaciones innecesarias del canal de *backup* usando un simulador con distintas topologías de red. Los resultados muestran que este porcentaje es mínimo.

Los mecanismos para establecer caminos disjuntos, la recuperación y reconfiguración de canales, están fuera del ámbito de esta tesis y existen soluciones bien conocidas a todos estos problemas que pueden ser introducidas en el esquema propuesto. Sin embargo, al final de este capítulo se esboza una forma simple de implementar este mecanismo.

## 6.2.- Esquema de detección de fallo

Como se ha comentado en el punto anterior, en una red en la que se dispone de varias rutas entre los nodos emisor y receptor (véase la figura 6.1) existen varios esquemas para proporcionar tolerancia a fallos en canales de tiempo real:

- **Múltiples canales activos (*Multiple active channels*):** Establecimiento de varios canales simultáneos, y envío de paquetes duplicados para que al receptor le llegue al menos uno. Este esquema funciona mientras el receptor reciba al menos uno de los paquetes replicados. Esta es una técnica costosa en recursos pero puede soportar fallos sin que se corte el servicio.
- **Activación en fallo (*reactive backup channels*):** Establecimiento de un canal primario, y en el caso de que falle, se restablecería la comunicación por otro canal secundario. El inconveniente es el tiempo de restablecimiento del canal secundario y la posibilidad de que no existan recursos disponibles cuando se active el canal de reserva.

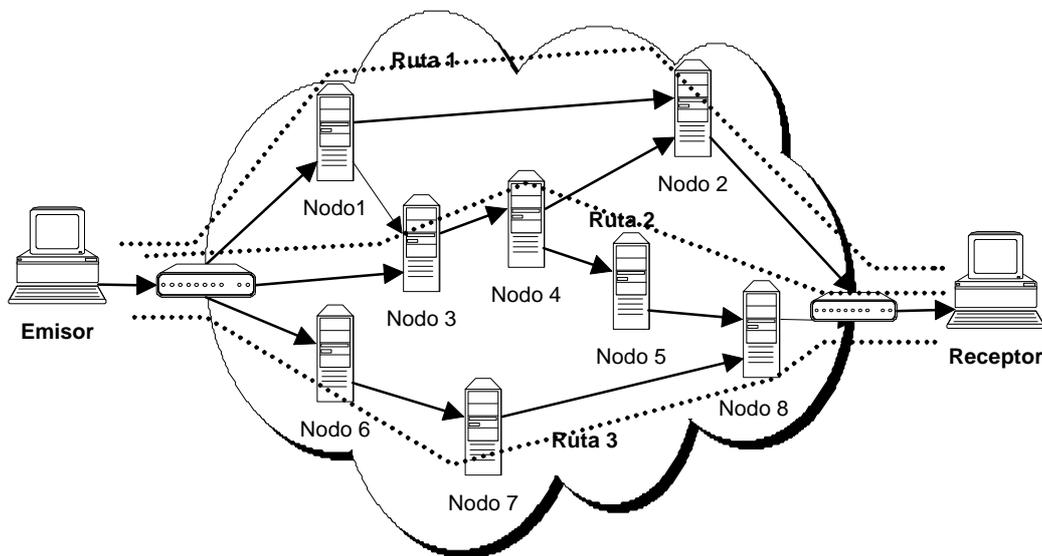


Figura 6.1 : Red con varias rutas.

Para evitar los inconvenientes del segundo esquema y que el establecimiento del canal secundario sea rápido se puede recurrir a reservar los recursos del canal secundario con anterioridad. Este último esquema requiere que se detecte el fallo del canal primario para establecer la comunicación por un canal secundario. Las propuestas existentes para la detección del fallo [Ramanathan92] normalmente se basan en el envío de paquetes del tipo “estoy vivo” (*heartbeat*) entre los nodos al emisor y receptor de un canal de tiempo real. Un problema importante comentado en [Han99] es la latencia de detección de fallo, especialmente en la técnica de detección emisor-receptor. Esta latencia puede ser un serio obstáculo en aplicar esta técnica, ya que provoca que los requerimientos de retraso en los canales de reserva sean muy exigentes para mantener un servicio ininterrumpido, y esto causa que las correspondientes reservas sean a veces inaceptables.

En este capítulo, se introduce un esquema de detección de fallos que permite una reserva de recursos más eficiente. El esquema propuesto se basa en “sospechar” un fallo cuando en el canal primario se produce un retraso próximo al retraso máximo, aunque siempre inferior a éste. Siempre que se sospecha este fallo, el canal de reserva se activa. Este esquema se denominará de activación anticipada del canal de reserva (*pre-reactive backup channel*). Esto da lugar, en ocasiones, a la detección de “falsos fallos” que provocan la activación innecesaria del canal de reserva. La ventaja es que la reserva que debe realizarse para el canal de reserva es significativamente menor que si se activara sólo ante “auténticos fallos”. En definitiva, se trata de sacrificar la exactitud en la detección con el objetivo de reducir la reserva de recursos. La única situación donde la inexactitud en la detección del fallo no se puede considerar aceptable sería con un tráfico a ráfagas, donde el emisor tiene periodos donde no transmite. Esto implicaría, o bien inyectar paquetes del tipo “estoy vivo” o modificar el tráfico para asegurar que el canal sea usado en estos periodos sin transmisión. Pero como se verá en la práctica, usando tráfico MPEG no existen estos periodos de inactividad debido al propio mecanismo de codificación MPEG.

Sea  $d_f$  el tiempo detección de fallo. Se asume que  $d_f$  es estrictamente menor que el máximo retraso en el canal primario  $d_A$ . El problema consiste en cómo escoger  $d_f$  para minimizar la reserva de recursos. En esta tesis se ofrecen algunas evaluaciones experimentales para diferentes tipos de tráfico.

Si se escoge que el tiempo de detección  $d_f$  sea igual al máximo retraso permitido en el canal primario, el retraso máximo experimentado por el primer paquete retransmitido por el canal de reserva cuando se produce un fallo en el canal primario se puede expresar como:

$$d_{total} = d_A + d_s + d_B \quad (1)$$

donde  $d_A$  y  $d_B$  son los retrasos máximos en los canales primario y de reserva respectivamente, y  $d_s$  el tiempo de establecimiento del canal de reserva desde que se descubre el fallo. Esto implica que para un retraso total  $d_{total}$  hay que establecer canales con unos retrasos bastante menores, lo que supone una reserva de recursos en la red mucho mayor (a menor retraso mayor consumo de recursos en la red).

De acuerdo a la técnica propuesta, la idea es escoger un límite en la detección de fallo  $d_f$  tal que  $d_f < d_A$  para permitir un valor de  $d_B$  mayor. Cuando el retraso de un paquete excede  $d_f$  se empieza a restablecer el canal de reserva, pero sin descartar el primario ya que podría ser un “falso fallo”. Así, el tiempo de retraso máximo se calcularía de la siguiente forma:

$$d_{total} = \max(d_A, d_f + d_e + d_B) \quad (2)$$

De esta forma,  $d_A$  puede ser  $d_{total}$  y el valor de  $d_B$  sería  $d_{total} - d_f - d_e$ . El hecho de que  $d_f$  sea siempre menor que  $d_A$  implica una importante reducción de la reserva de recursos en el canal de reserva. Además, los recursos asignados al canal de reserva los podrían utilizar el *router* mientras no se usen, para la transmisión de otro tipo paquetes no prioritarios, con lo que realmente no se están desperdiciando recursos. El único momento donde se desperdician recursos es cuando se establece el canal B cuando se detecta un fallo. Esto plantea los siguientes temas:

- Cómo calcular el valor de  $d_f$ . Si este valor es alto, el valor  $d_B$  tendrá que ser bajo con lo que habrá una reserva elevada de recursos, y si el valor  $d_f$  es bajo provocará que se establezca el canal de reserva con demasiada frecuencia con el consiguiente derroche de recursos ante falsos fallos.
- Cómo calcular el coste de establecer el canal de reserva.

### 6.3.- Redes de prueba y cálculo de la reserva

En este punto se describen las redes utilizadas para probar el nuevo esquema y el método usado para calcular la reserva óptima. Como se ha descrito en capítulos anteriores, el retraso emisor-receptor se puede calcular usando dos ecuaciones. Las especificadas por IETF en el servicio garantizado que usan el modelo de tráfico *token bucket* ( $b, r, p$ ) y las desarrolladas por *Parekh/ Gallagher* que usan el modelo *leaky bucket*.

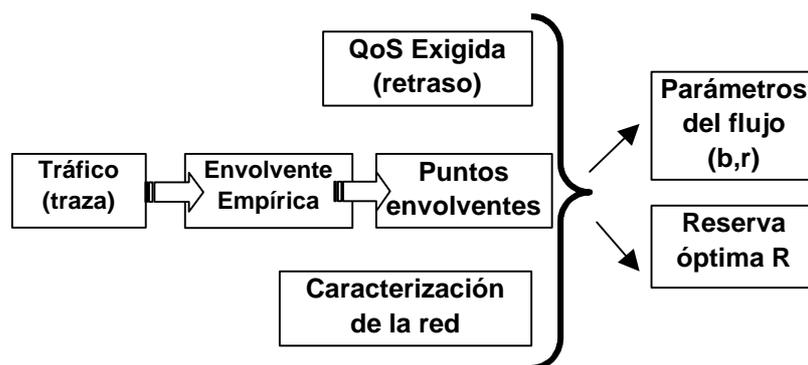


Figura 6.2: Proceso para obtener la reserva óptima.

Resumiendo, el proceso es como sigue (véase figura 6.2): del tráfico se obtiene la envolvente empírica y los puntos envolventes. Con los requerimientos de calidad de

servicio (retraso máximo) y la caracterización de la red, se obtienen los parámetros  $s$  y  $r$  que optimizan los recursos de red.

La figura 6.3 describe las redes usadas en las pruebas. La Red 1 es una red IP típica con 3 nodos y anchos de banda del orden de 50 Mb. La Red 2 tiene 9 nodos y es un ejemplo de una red con muchos saltos, y la Red 3 es una red similar a ATM con un tamaño de paquete fijo (53 bytes) y grandes anchos de banda. Los planificadores de los nodos son el conocido WFQ (*Weighted Fair Queuing*) [Demers89]. Los parámetros de la red están en la tabla 6.1 donde  $M$  es el tamaño máximo del paquete y  $m$  el tamaño mínimo (*minimum policed unit*)

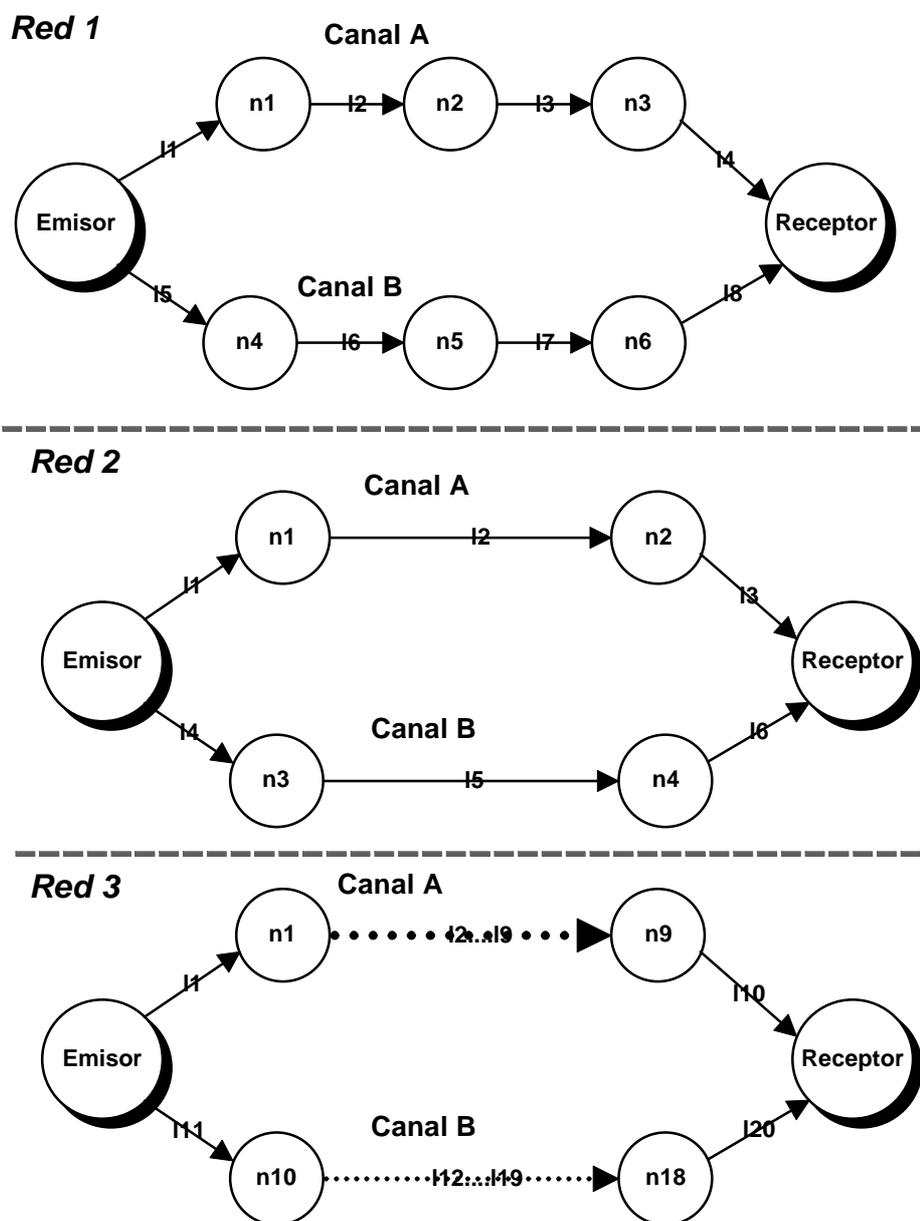


Figura 6.3: Redes ejemplo.

Red 1			Red 2			Red 3		
Enlace	A.B. (Mbps)	MTU (bits)	Enlace	A.B. (Mbps)	MTU (bits)	Enlace	A.B. (Mbps)	MTU (bits)
11,15	50	10000	11,15,19,111,115,119	50	25000	11,14	100	428
12,16	35	12000	12,16,110,112,16,120	100	20000	12,15	52	428
13,17	100	15000	13,17,113,117	52	15000	13,16	155	428
14,18	50	20000	14,18,114,118	50	20000			
M = 1Kbyte m = 1000 bits			M = 1Kbyte m = 500 bits			M = m = 428 bits		

Tabla 6.1: Parámetros de la redes ejemplo.

A partir de estas redes es fácil obtener los valores de  $C_{tot}$  y  $D_{tot}$ , como por ejemplo para la Red 1:

$$D_{tot} = \sum D_i = \sum \frac{MTU_i}{B_i} = 0.00109 \quad C_{tot} = \sum C_i = \sum M = 32768$$

El retraso total en la red es la suma del retraso en cola ( $Q_{delay}$ ) y la latencia mínima del camino ( $l = 0.0006$  s.). Sin pérdida de generalidad, para todos los casos el tiempo de establecimiento del canal de backup  $d_s$  será considerado 0. En el último punto de este capítulo se verá que este valor es muy pequeño y no afecta a los resultados.

## 6.4.- Evaluación analítica del ahorro de recursos

### 6.4.1.- Introducción

El objetivo de esta sección es comparar la reserva de recursos del esquema propuesto con los otros esquemas introducidos en la sección 6.2. Estos resultados serán usados en la próxima sección, para evaluar la bondad de este nuevo esquema simulando los mismos escenarios. Para esta evaluación se utilizan algunos tráfico MPEG descritos en el apéndice 1. Los tres esquemas pueden ser comparados fácilmente usando el método descrito en la sección 6.3 para obtener la reserva óptima en la red. Sólo falta decidir una cosa: el retraso total  $d_{total}$  y los retrasos máximos del canal primario y secundario ( $d_A, d_B$ ).

- **A&B** (Copia múltiple): El emisor envía siempre por los canales A y B simultáneamente. En este esquema, el tiempo de retraso máximo de los canales es igual al requerido  $d_{total}$ . Este esquema es costoso porque implica que el canal B siempre esté usado.
- **A|B** (Activa en fallo): Este esquema espera a detectar un fallo en el canal A para emitir por el B. La detección del fallo se producirá cuando un paquete se retrase más del valor  $d_{total}/2$ . Por lo tanto, para calcular los retrasos en cada canal se dividirá  $d_{total}$  por 2. Este esquema no utiliza el canal B hasta el

momento en que se produce un error. Aunque se tengan reservados los recursos del canal, los *routers* pueden utilizar el ancho de banda reservado para transmitir tráfico no prioritario, con lo que no se desperdician los recursos asignados.

- **A<sup>^</sup>B** (Activación anticipada): El esquema propuesto da a A un retraso menos exigente que es igual a  $d_{total}$  y a B la mitad de retraso total, con lo que se asume que  $d_f$  es también la mitad del retraso. Al igual que el esquema A|B, el canal B no se utiliza hasta que el retraso sea mayor que  $d_f$ . Pero aquí se pueden producir falsos fallos ya que puede que llegue el paquete esperado por el canal A, con lo que realmente no se ha producido ningún fallo. A partir del tráfico se puede calcular aproximadamente que porción del tráfico provocará que se empiece a restablecer el canal B. Se denominará a este valor tasa de falso fallo: TFF (FFR: *False Fault Rate*).

En todas las evaluaciones se selecciona un valor reducido de  $d_{total}$  con el objetivo de reservar el máximo de recursos de la red. Usando este valor base se seleccionan dos valores múltiplo.

A partir de la descripción del tráfico, es posible calcular la tasa de falso fallos (TFF). Este valor es una estimación de cuántas veces el canal de reserva será activado debido a un falso fallo. Esta activación del canal de reserva se producirá normalmente cuando hay un pico en la transmisión y el retraso de los paquetes se dispara. Por tanto, el objetivo es obtener la porción de tráfico que provoca un retraso mayor que  $d_{total} / 2$ . Para calcular este valor se puede utilizar la siguiente aproximación: si se eliminan los picos de un tráfico, es decir, las tramas de mayor tamaño, se obtiene un nuevo tráfico “reducido”, menos variable. Este tráfico reducido da menos retraso que el original al tener pico menores y ser menos variable. En la figura 6.4 se muestra el tráfico SOCCER1500<sup>14</sup> original, y el tráfico reducido en 1% en la figura 6.5, que es el resultado de eliminar las 150 tramas de mayor tamaño. Este nuevo tráfico tiene una tasa pico de 2464167b/s y una tasa media de 584922b/s, que son bastante menores que el tráfico original.

---

<sup>14</sup> En este capítulo se han cogido extractos de los tráficos de Rose que consisten en las primeras 1500 tramas para que las simulaciones no fuesen excesivamente costosas. En el apéndice 1 se describen con detalle estos tráficos.

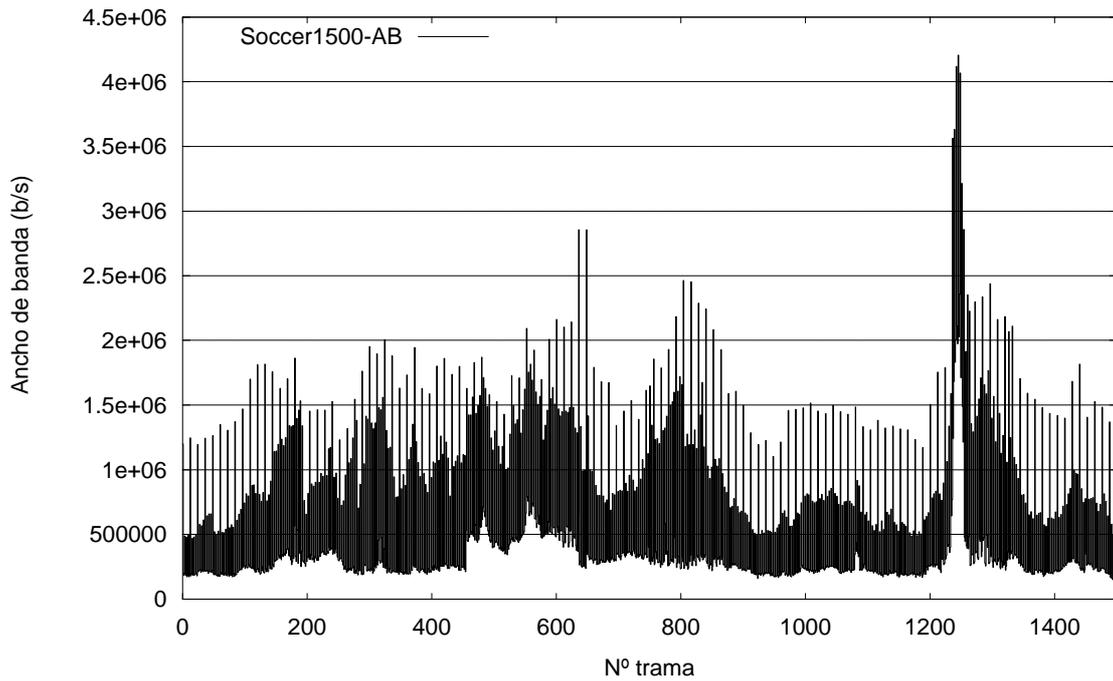


Figura 6.4 : Tráfico SOCCER1500 origen.

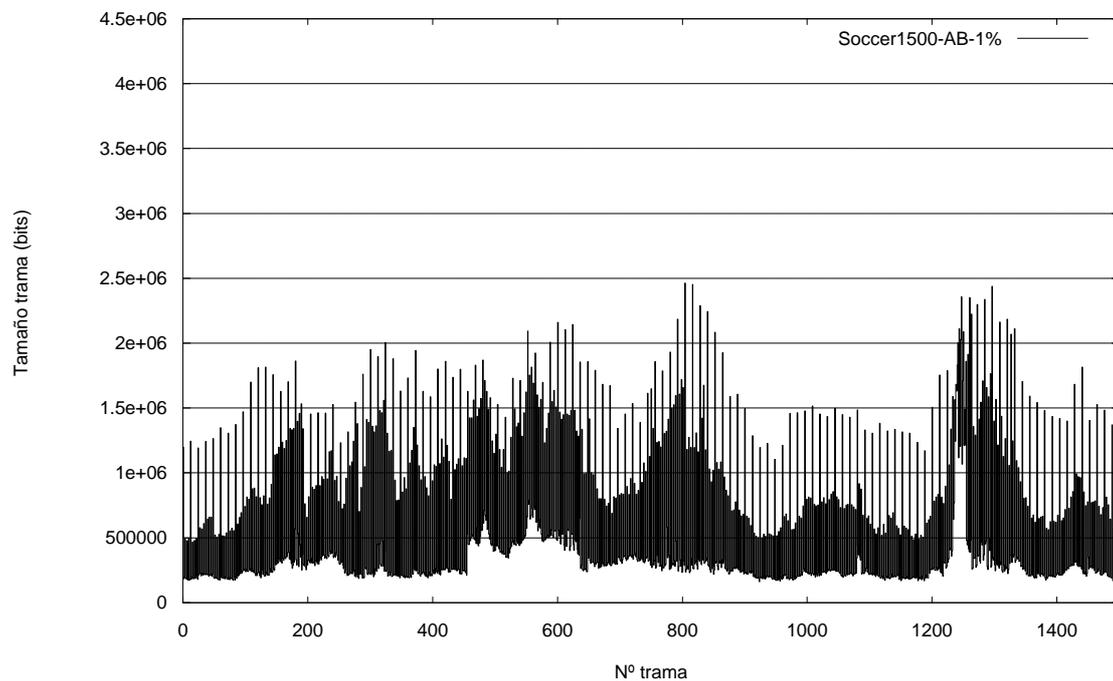


Figura 6.5 : Tráfico SOCCER1500 reducido.

De esta forma, la tasa de falso fallos (TFF) es la tasa de reducción del tráfico que produce un retraso de  $d_{total}/2$ . Por ejemplo, usando la red 1 y fijando el retraso en 0.02s, si el tráfico SOCCER1500 se reduce en 1% entonces el nuevo tráfico da un retraso menor  $d_x$  de 0.016s usando los mismo parámetros de red. Por lo tanto, el objetivo es obtener la tasa

de reducción que proporciona un retraso de  $d_{total} / 2$ . Esta porción del tráfico es la que provoca la diferencia de retraso entre  $d_{total}$  y  $d_x$ . Para calcular el retraso, el tráfico se reduce primero en una tasa  $r$  y después se obtiene los parámetros  $b$  y  $r$  con el método visto. Entonces, con estos parámetros y la reserva con el tráfico original se calcula el retraso usando las ecuaciones de retraso de IETF/RSVP (9) y (10) del capítulo 2. De forma iterativa se obtiene una aproximación a la tasa de falsos fallos.

### 6.4.2.- Evaluación para la red 1

La tabla 6.2 muestra los resultados para todos los tráficos en la red 1. Se usan tres retrasos para cada comparación (0.1s, 0.05s y 0.02s). Los retrasos en los canales A y B ( $d_A$  y  $d_B$ ) y sus reservas correspondientes se obtienen a partir de  $d_{total}$ . Por ejemplo, con un retraso  $d_{total} = 0.1s$  el retraso máximo  $d_A$  y  $d_B$  para el esquema **A&B** es de 0.1s. Para el esquema **A|B** el retraso será 0.05s en ambos canales. Para **A^B** el retraso será 0.1s para el canal A y 0.05s para el canal B.

Con estos retrasos se obtienen las reservas en cada canal  $R_A$  y  $R_B$ . Con la suma de las reservas de los dos canales se obtiene la reserva total  $R_{total}$ . La fila “% A^B” indica el ahorro de recursos obtenido por el esquema **A^B** contemplando ambos canales o sólo el primario.

	A&B			A B			A^B			
$d_{total}(s)$	0.1	0.05	0.02	0.1	0.05	0.02	0.1	0.05	0.02	
$d_A(s)$	0.1	0.05	0.02	0.05	0.025	0.01	0.1	0.05	0.02	
$d_B(s)$	0.1	0.05	0.02	0.05	0.025	0.01	0.05	0.025	0.01	
<b>LAMBS1500</b>										
$R_A(Mb/s)$	0.87	1.36	2.05	1.36	1.89	4.6	0.87	1.36	2.16	
$R_B(Mb/s)$	0.87	1.36	2.05	1.36	1.89	4.6	1.36	1.89	4.6	
$R_{total}(Mb/s)$	<b>1.74</b>	<b>2.72</b>	<b>4.1</b>	<b>2.72</b>	<b>3.78</b>	<b>9.2</b>	<b>2.23</b>	<b>3.25</b>	<b>6.76</b>	
% A^B	-/50	-/50	-/50	18/36	14/28	27/53				
							<b>TFF</b>	0.055	0.023	0.04
<b>SOCCER1500</b>										
$R_A(Mb/s)$	2.43	2.74	3.53	2.74	3.25	4.25	2.43	2.74	3.53	
$R_B(Mb/s)$	2.43	2.74	3.53	2.74	3.25	4.25	2.74	3.25	4.25	
$R_{total}(Mb/s)$	<b>4.86</b>	<b>5.48</b>	<b>7.06</b>	<b>5.48</b>	<b>6.5</b>	<b>8.5</b>	<b>5.17</b>	<b>5.99</b>	<b>7.78</b>	
% A^B	-/50	-/50	-/50	5.6/11	7.8/16	8.4/17				
							<b>TFF</b>	0.004	0.004	0.004
<b>NEWS1500</b>										
$R_A(Mb/s)$	0.57	0.89	2.17	0.89	1.73	4.6	0.57	0.89	2.17	
$R_B(Mb/s)$	0.57	0.89	2.17	0.89	1.73	4.6	0.89	1.73	4.6	
$R_{total}(Mb/s)$	<b>1.14</b>	<b>1.78</b>	<b>4.34</b>	<b>1.78</b>	<b>3.46</b>	<b>9.2</b>	<b>1.46</b>	<b>2.62</b>	<b>6.77</b>	
% A^B	-/50	-/50	-/50	18/28	24/49	26/53				
							<b>TFF</b>	0.084	0.039	0.057

Tabla 6.2: Reserva para la red 1.

En general, el ahorro de recursos es mayor cuando el retraso exigido es menor. Así, en el tráfico LAMBS1500 para  $d_{total} = 0.02s$  se ahorra cerca de 2.5Mb/s de ancho de banda sobre el esquema A|B (27% de ahorro), pero para 0.05s el ahorro es de sólo 0.53Mb/s (14%). En el tráfico SOCCER1500, el ahorro es de 0.7Mb/s sobre A|B (8.4%) para el retraso más exigente de 0.02s. En el tráfico NEWS1500, el ahorro es de 0.84Mb/s sobre A|B (24%) para un retraso de 0.05s. Estos resultados son para cada enlace en el camino, por lo que para el primer ejemplo el ahorro total de recursos en la red es de 10Mb/s. Asumiendo que los recursos del canal B pueden ser usados por el router cuando el canal no está en uso (esto es, considerando sólo los recursos del canal A), los resultados son substancialmente mejores. Así, en el tráfico NEWS1500, el ahorro es de 0.84Mb sobre A|B (49% de ahorro) y 0.89Mb sobre A&B (50%) para  $d_{total} = 0.05s$ .

Como es lógico, la mejora sobre el esquema A&B es siempre del 50% dado que al ser necesarios en este esquema los dos canales, la reserva es el doble que en el esquema propuesto.

Las tasas de falsos fallos estimadas son bastantes pequeñas para todos los retrasos calculados. Se puede hacer una estimación del coste que supone los falsos fallos en el uso del canal B. Se obtiene para ello una reserva equivalente que es igual a:

$$R_{eq} = TFF \times R_B$$

Para el ejemplo anterior del tráfico NEWS1500, la reserva equivalente es  $0.039 \times 1.73 = 0.067$  que sumando a la reserva del canal A da una reserva total de 0.96, que es aún menor que en los otros dos esquemas.

La ganancia de recursos depende principalmente de la diferencia entre la reserva obtenida para  $d_{total}$  (valor  $R$ ) y la reserva obtenida para  $d_{total}/2$  (valor  $R'$ ). La figura 6.5 muestra gráficamente esta diferencia para el tráfico LAMBS1500 para un retraso exigido entre 0 a 1s. El mayor ahorro se produce para retrasos muy pequeños (0.01s), después decae un poco para tener luego un pico alrededor de 0.2s. De 0.2 a 0.4s el ahorro decae rápidamente para luego incrementar muy despacio. Para el tráfico SOCCER1500 (véase figura 6.6) las curvas son muy diferentes. La ganancia es alta para retrasos muy pequeños, pero decrece muy rápidamente hasta 0.07s para luego crecer linealmente. El tráfico NEWS1500 (véase la figura 6.7) es muy similar al tráfico LAMBS1500. La figura 6.8 muestra la ganancia total para los tres tipos de tráfico en la red 1.

En general, se puede ver que la ganancia es muy alta para retrasos pequeños y luego cae muy rápidamente. Esta caída es debido a la transición entre las dos ecuaciones de retraso (9) y (10). Para retrasos muy pequeños se utiliza la ecuación (10) donde la reserva

decae muy rápidamente y casi linealmente (véase en la figura 6.5 la curva  $R'$  entre 0.02s y 0.05s). Pero cuando el retraso pasa esta frontera se utiliza la ecuación (9) la cual proporciona una curva que decae al principio exponencialmente para luego decaer lentamente (véase en la figura 6.5 la curva  $R'$  entre 0.05 y 1s). Este patrón se repite en todos los tráficos y redes. Es la diferencia entre los retrasos que provocan el cambio de ecuación la que configura las curvas de ahorro de recursos.

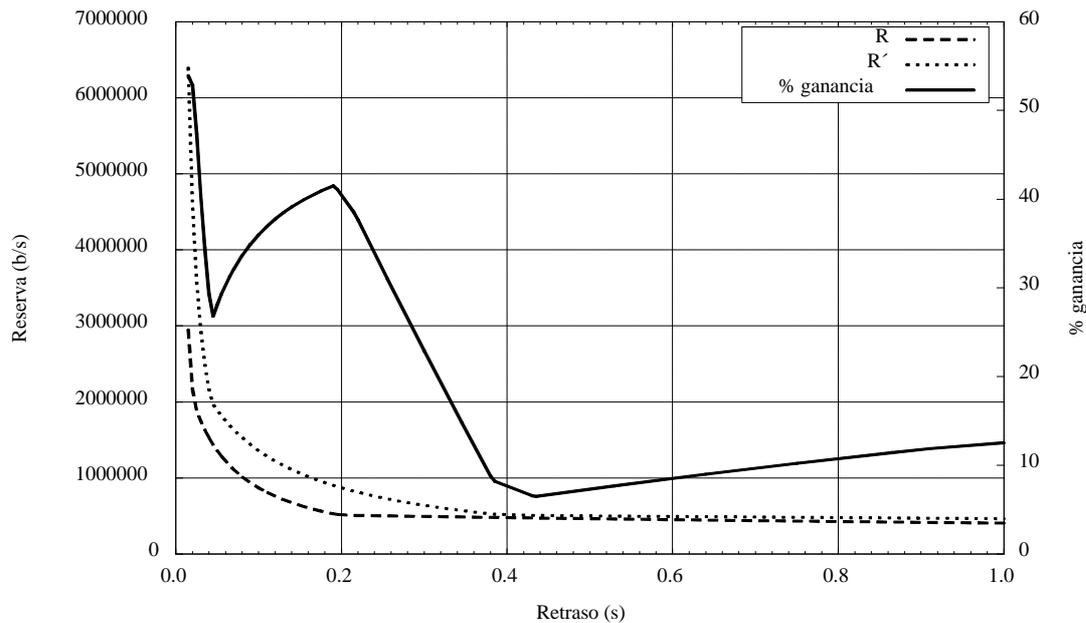


Figura 6.5: Reserva y ganancia para tráfico LAMBS1500 en red 1.

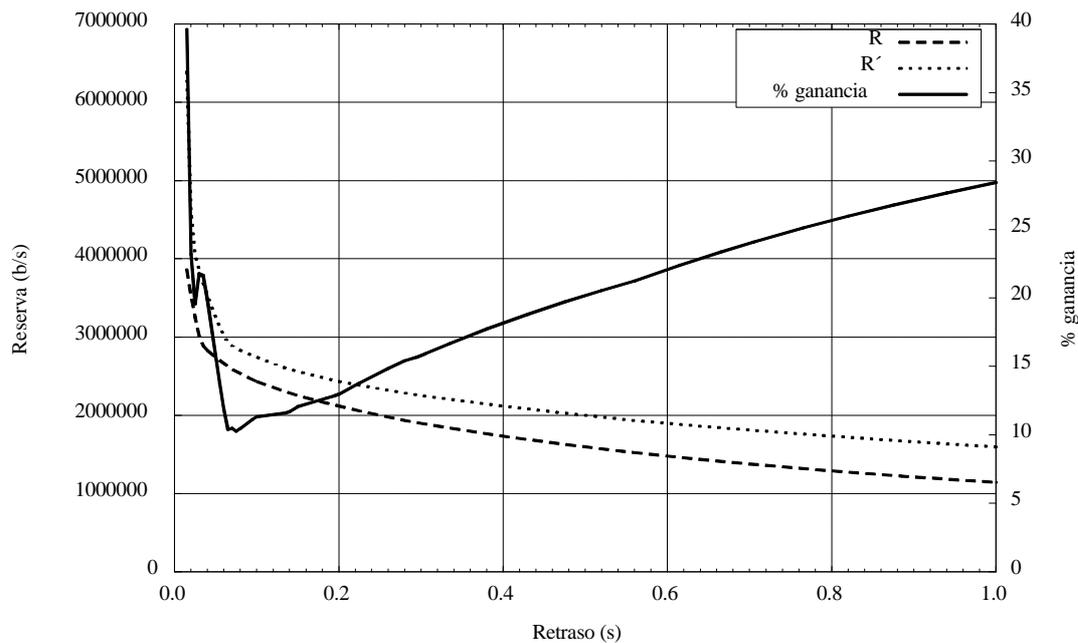


Figura 6.6: Reserva y ganancia para tráfico SOCCER1500 en red 1.

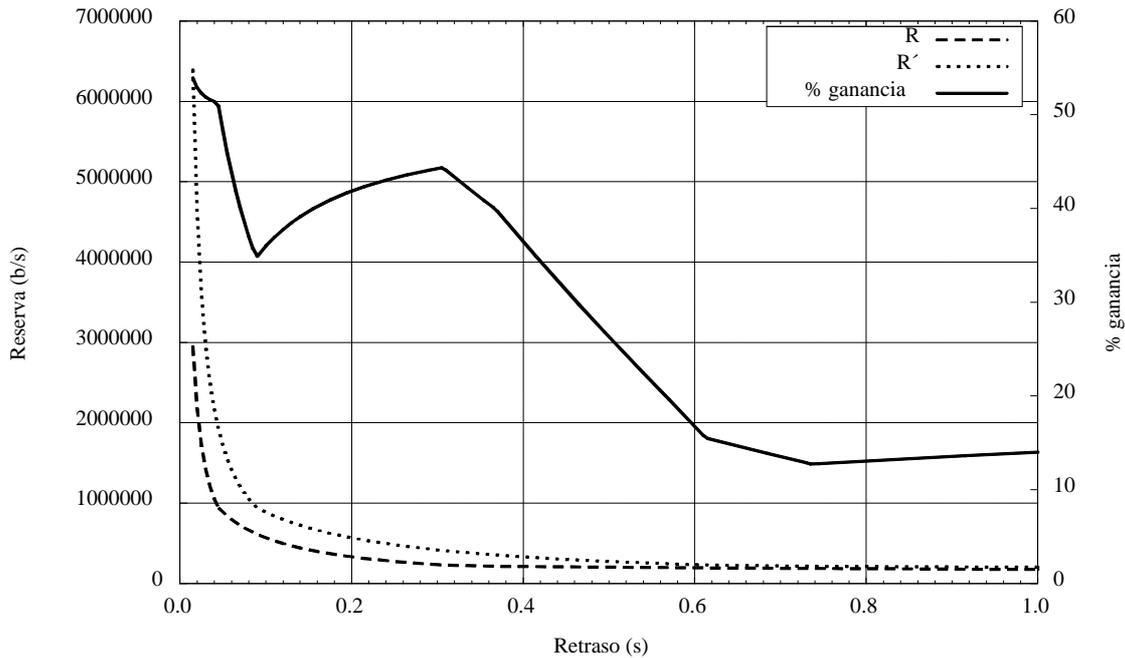


Figura 6.7: Reserva y ganancia para tráfico NEWS1500 en red 1.

#### 6.4.3.- Evaluación para la red 2

En esta red se evalúa la reserva en una red con muchos nodos. La tabla 6.3 muestra los resultados usando 3 retrasos diferentes ( $d_{total}$ ) que debido a la longitud de la red son bastante mayores que en la red anterior. La figura 6.9 muestra la ganancia de 0 a 1s, que en general su forma es muy similar a la Red 1. También se aprecia que la ganancia es un poco mayor que en la red anterior. Esto es debido a que al tener más nodos la diferencia entre las ecuaciones  $R$  y  $R'$  se hace mayor.

#### 6.4.4.- Evaluación para la red 3

En esta red, debido al elevado ancho de banda de los enlaces y el pequeño tamaño del paquete los retrasos son muy pequeños. La tabla 6.4 muestra los resultados. En este caso las curvas de ganancia en porcentaje difieren de las redes anteriores (véase la figura 6.10). Tiene una tasa pico en valores muy pequeños de retraso y decae rápidamente a aproximadamente 0 en 0.005 segundos para a continuación subir exponencialmente, menos en el tráfico SOCCER1500. El ahorro de recursos, en general, es menor que en las redes anteriores.

	A&B			A B			A^B			
$d_{total}(s)$	<b>0.4</b>	<b>0.2</b>	<b>0.1</b>	<b>0.4</b>	<b>0.2</b>	<b>0.1</b>	<b>0.4</b>	<b>0.2</b>	<b>0.1</b>	
$d_A(s)$	0.4	0.2	0.1	0.2	0.1	0.05	0.4	0.2	0.1	
$d_B(s)$	0.4	0.2	0.1	0.2	0.1	0.05	0.2	0.1	0.05	
<b>LAMBS1500</b>										
$R_A(Mb/s)$	0.49	0.72	1.24	0.72	1.24	1.96	0.49	0.72	1.24	
$R_B(Mb/s)$	0.49	0.72	1.24	0.72	1.24	1.96	0.72	1.24	1.96	
$R_{total}(Mb/s)$	<b>0.98</b>	<b>1.44</b>	<b>2.48</b>	<b>1.44</b>	<b>2.48</b>	<b>3.92</b>	<b>1.21</b>	<b>1.96</b>	<b>3.2</b>	
% A^B	-/50	-/50	-/50	16/32	21/42	18/37				
							<b>FFR</b>	0.078	0.061	0.06
<b>SOCCER1500</b>										
$R_A(Mb/s)$	1.78	2.18	2.54	2.18	2.54	2.97	1.78	2.18	2.54	
$R_B(Mb/s)$	1.78	2.18	2.54	2.18	2.54	2.97	2.18	2.54	2.97	
$R_{total}(Mb/s)$	<b>3.56</b>	<b>4.36</b>	<b>5.08</b>	<b>4.36</b>	<b>5.08</b>	<b>5.94</b>	<b>3.96</b>	<b>4.72</b>	<b>5.51</b>	
% A^B	-/50	-/50	-/50	9/18	7/14	7/14				
							<b>FFR</b>	0.078	0.059	0.047
<b>NEWS1500</b>										
$R_A(Mb/s)$	0.29	0.54	0.94	0.54	0.94	1.94	0.29	0.54	0.94	
$R_B(Mb/s)$	0.29	0.54	0.94	0.54	0.94	1.94	0.54	0.94	1.94	
$R_{total}(Mb/s)$	<b>0.58</b>	<b>1.08</b>	<b>1.88</b>	<b>1.08</b>	<b>1.88</b>	<b>3.88</b>	<b>0.83</b>	<b>1.48</b>	<b>2.88</b>	
% A^B	-/50	-/50	-/50	23/46	21/43	26/52				
							<b>FFR</b>	0.084	0.1	0.12

Tabla 6.3: Reserva para la red 2.

	A&B			A B			A^B			
$d_{total}(s)$	<b>0.1</b>	<b>0.01</b>	<b>0.001</b>	<b>0.1</b>	<b>0.01</b>	<b>0.001</b>	<b>0.1</b>	<b>0.01</b>	<b>0.001</b>	
$d_A(s)$	0.1	0.01	0.001	0.05	0.005	0.0005	0.1	0.01	0.001	
$d_B(s)$	0.1	0.01	0.001	0.05	0.005	0.0005	0.05	0.005	0.0005	
<b>LAMBS15000</b>										
$R_A(Mb/s)$	0.64	1.78	2.18	0.99	1.98	3.53	0.64	1.78	2.18	
$R_B(Mb/s)$	0.64	1.78	2.18	0.99	1.98	3.53	0.99	1.98	3.53	
$R_{total}(Mb/s)$	<b>1.28</b>	<b>3.56</b>	<b>4.36</b>	<b>1.98</b>	<b>3.96</b>	<b>7.06</b>	<b>1.63</b>	<b>3.76</b>	<b>5.71</b>	
% A^B	-/50	-/50	-/50	18/35	5/10	19/38				
							<b>FFR</b>	0.023	0.01	0.04
<b>SOCCER1500</b>										
$R_A(Mb/s)$	2.38	3.53	4.31	2.64	3.92	4.36	2.38	3.53	4.31	
$R_B(Mb/s)$	2.38	3.53	4.31	2.64	3.92	4.36	2.64	3.94	4.36	
$R_{total}(Mb/s)$	<b>4.76</b>	<b>7.06</b>	<b>8.62</b>	<b>5.28</b>	<b>7.84</b>	<b>8.72</b>	<b>5.02</b>	<b>7.47</b>	<b>8.67</b>	
% A^B	-/50	-/50	-/50	5/10	5/10	0.6/1				
							<b>FFR</b>	0.004	0.003	0.002
<b>NEWS1500</b>										
$R_A(Mb/s)$	0.34	0.95	1.74	0.53	1.05	3.53	0.34	0.95	1.74	
$R_B(Mb/s)$	0.34	0.95	1.74	0.53	1.05	3.53	0.53	1.05	3.53	
$R_{total}(Mb/s)$	<b>0.68</b>	<b>1.9</b>	<b>3.48</b>	<b>1.06</b>	<b>2.1</b>	<b>7.06</b>	<b>0.87</b>	<b>2</b>	<b>5.27</b>	
% A^B	-/50	-/50	-/50	18/36	5/10	25/51				
							<b>FFR</b>	0.08	0.084	0.12

Tabla 6.4: Reserva para la red 3.

### 6.4.5.- Conclusiones

Se han agrupado las curvas de ganancia en las siguientes páginas para que se puedan comparar fácilmente (véanse las figuras 6.8, 6.9 y 6.10).

En resumen, el esquema propuesto A<sup>^</sup>B reserva un total de recursos que está entre los otros dos esquemas. En general, los mayores ahorros se obtienen con el tráfico LAMBS1500, pero con una tasa de falsos fallos de cercana al 10%. Esto es razonable debido a que es un tráfico con un patrón bastante regular. En el otro extremo, el tráfico SOCCER1500 tiene menores ganancias, debido a su alta variabilidad. En cambio, tiene la mejor TFF. La reserva de recursos está entre 5 y 30% sobre el protocolo A|B, y considerando sólo el canal primario el ahorro de recursos está entre 10 y 50% sobre los esquemas A&B y A|B.

Por redes, no hay mucha diferencia entre la red 1 y red 2, teniendo esta última un ahorro de recursos mayor. En cambio en la red 3 es diferente la reserva y menor el ahorro que en las redes anteriores. Esto es debido al pequeño y fijo tamaño del paquete que hacen que las ecuaciones de retrasos obtengan valores más precisos y continuos.

## 6.5.- Resultados de Simulaciones

### 6.5.1.- Introducción

El objetivo de esta sección es validar el nuevo esquema introducido evaluando cuántas veces el canal de reserva se activa debido a un falso fallo. Se mostrará experimentalmente que esta tasa (TFF) es muy baja y menor que la obtenida analíticamente. Esto es así, principalmente debido al hecho de que las ecuaciones usadas para garantizar el retraso máximo de un paquete dan un límite muy pesimista del tráfico, que sólo ocurre cuando la red está totalmente cargada. Normalmente, los paquetes llegan mucho antes que sus retrasos máximos.

Para este propósito, se ha desarrollado el simulador de red **RTNetSim** (véase el apéndice A.2). El tráfico es introducido en la red por medio del esquema *token bucket* o *leaky bucket* y se utiliza en ambos casos un planificador WFQ. Uno de los problemas que surgen en cualquier simulación que compare esquemas es cómo cargar la red. Para resolver este problema, se crean 10 canales en cada nodo con un índice de carga que varía del 0% (sin carga) al 100% (carga total). El índice de carga es el porcentaje de uso del ancho de banda del enlace no reservado.

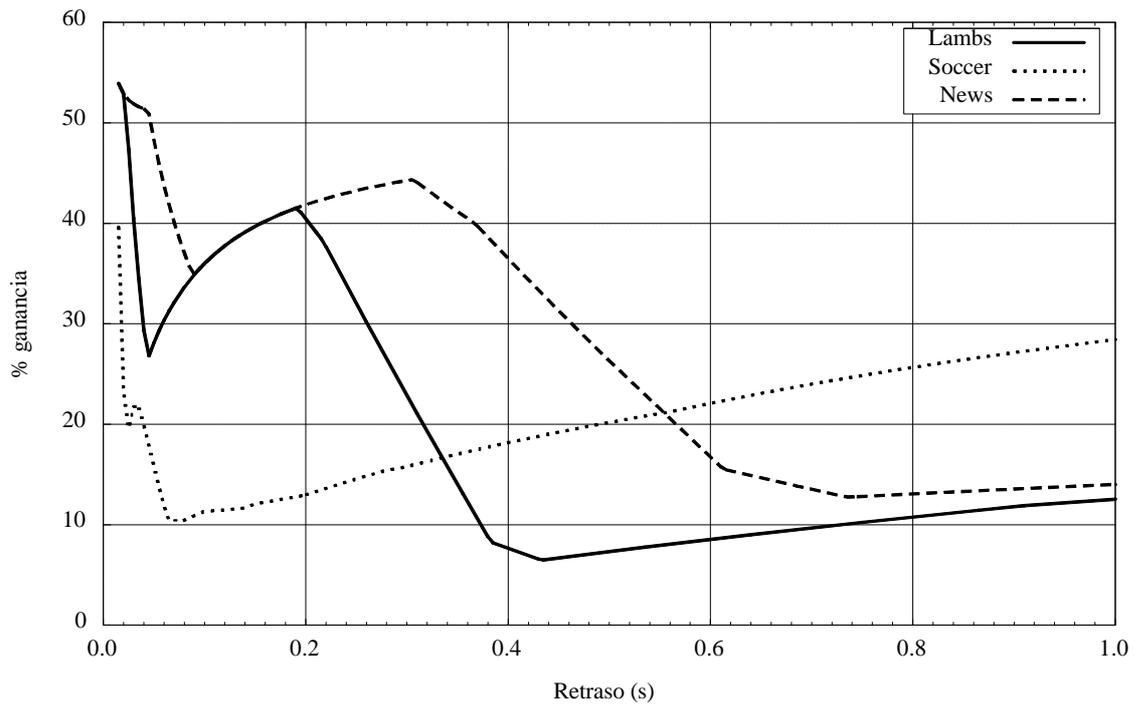


Figura 6.8: Ganancia para todos los tráficos en la red 1.

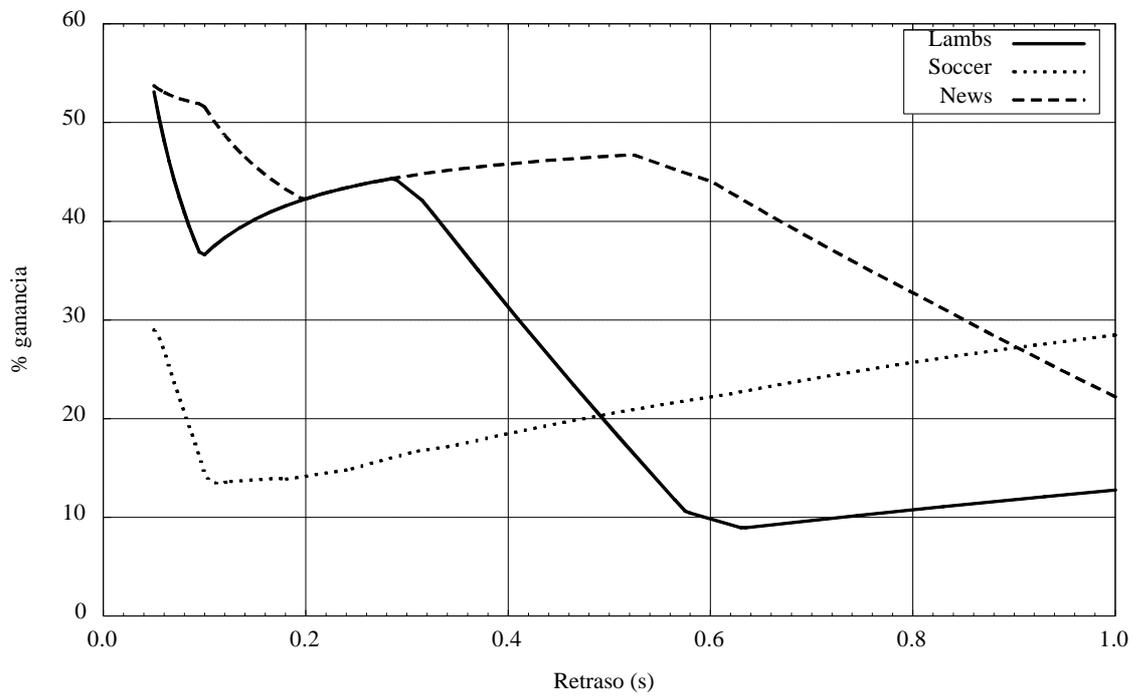


Figura 6.9: Ganancia para todos los tráficos en la red 2.

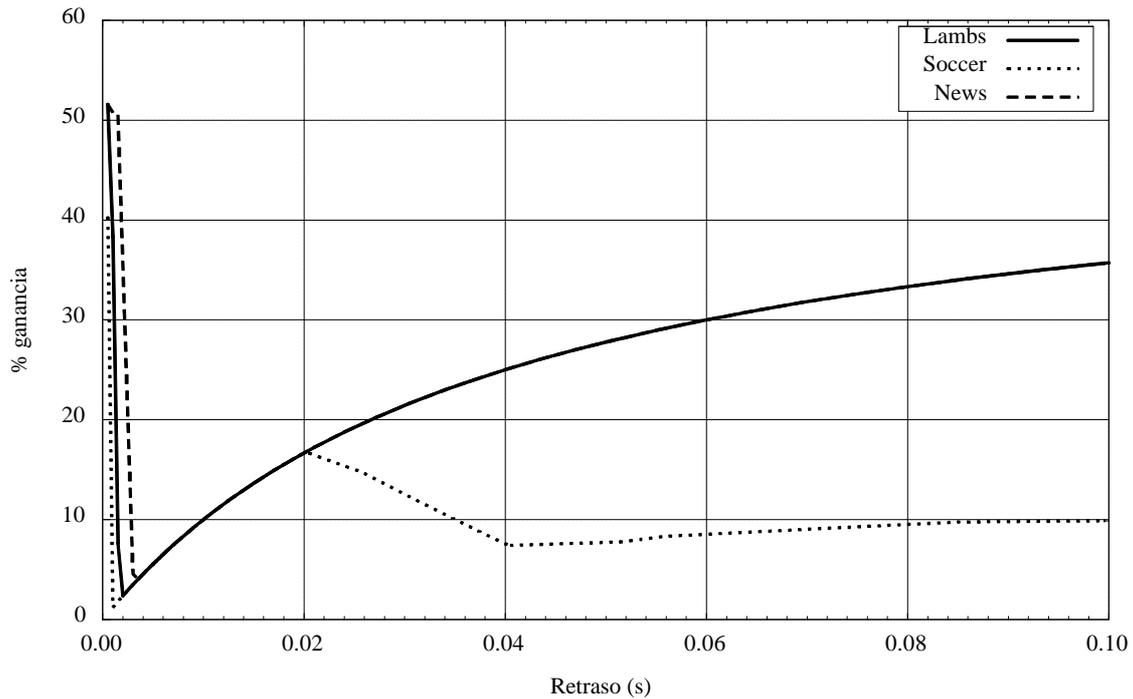


Figura 6.10: Ganancia para todos los tráficos en la red 3.

Aunque la reserva de ancho de banda y los parámetros de flujo son los mismos para los modelos de tráfico *leaky bucket* y *token bucket*, el resultado de las simulaciones obviamente difiere. En general, esto es debido a la forma en que el emisor introduce los paquetes en la red. Primero se van a presentar los resultados para *token bucket*.

Para ver que la mayoría de los paquetes llegan mucho antes que el retraso máximo, la figura 6.11 representa las funciones de densidad de llegada de paquetes en las simulaciones realizadas para la red 1 usando un retraso de 0.01s con el tráfico SOCCER1500 para distintos niveles de carga. En la figura 6.12 están los mismos resultados para la red 2 pero usando un retraso mayor de 0.05s. Se ve claramente que todos los paquetes llegan mucho antes que sus retrasos máximos y que cuanto más cargada esté la red, los paquetes se retrasan más. Estos dos hechos obvios son la base de la eficiencia del esquema propuesto. Por ejemplo, en la red 1 con carga total, el máximo retraso es de 0.024s que está por debajo del límite (0.025s). En la red 2 sólo con carga total, el retraso de algunos paquetes está por encima del límite que implica activar el canal de backup: 0.025s. Esto implica que **todos** los nodos estén totalmente cargados (una condición muy improbable).

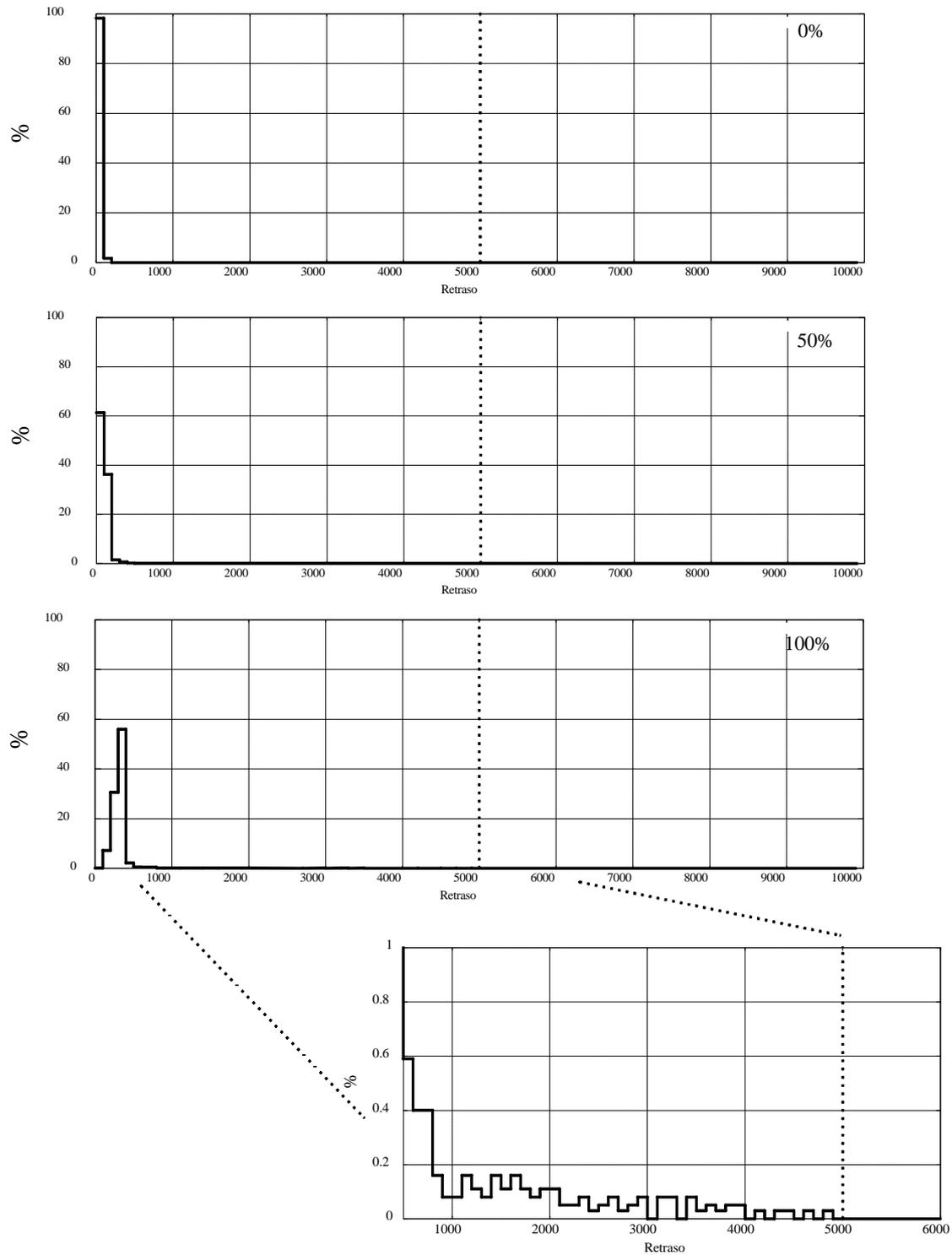


Figura 6.11: Densidad de llegada de paquetes para el tráfico SOCCER1500 en la red 1 para un retraso de 0.01s. (Los valores de retraso en el gráfico están en **ms**). Se representan tres niveles de carga (0%: sin carga, 50% media carga y 100%: carga total). La última gráfica es una porción ampliada de la carga total entre 500 y 6000**ms**.

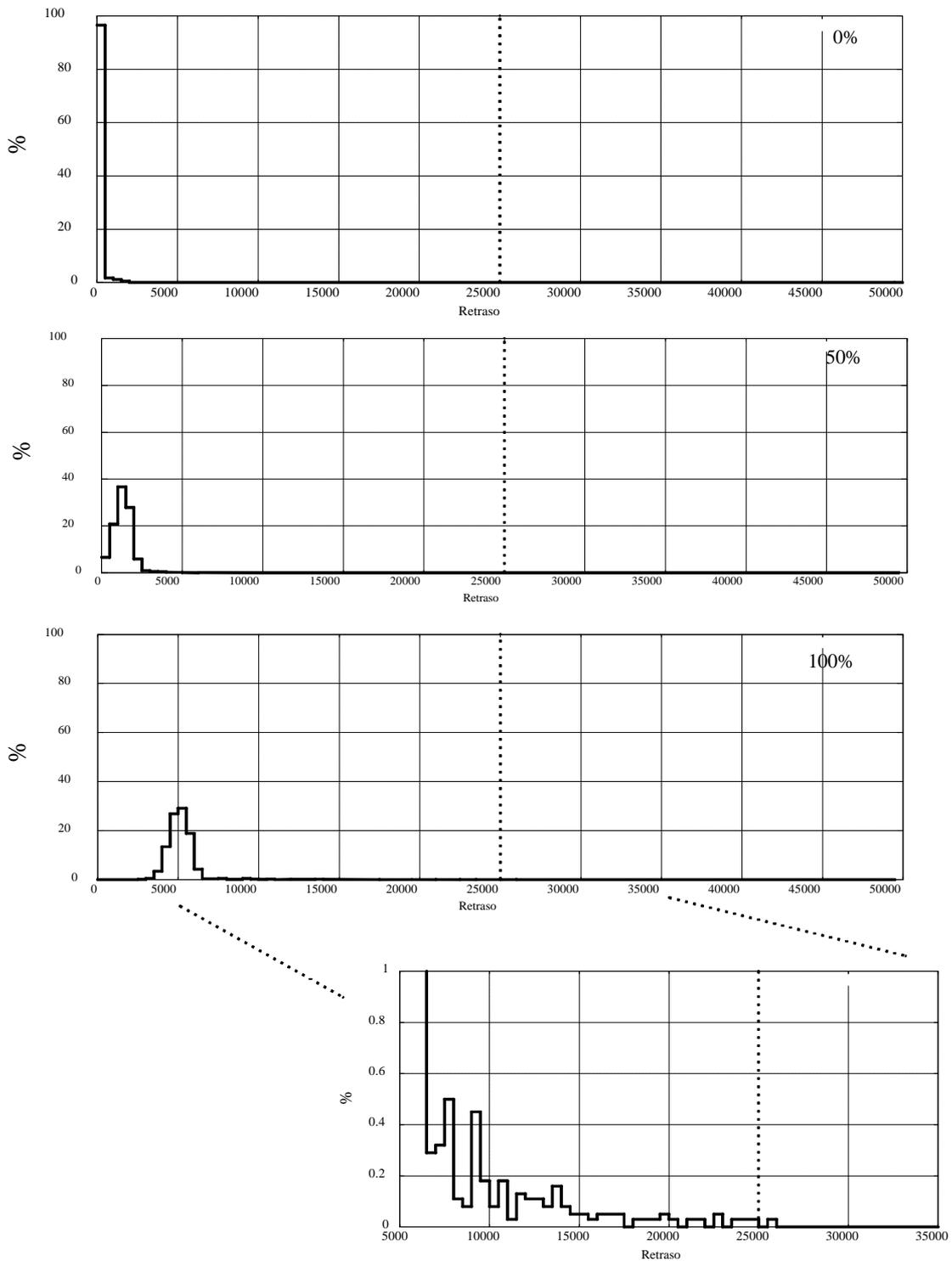


Figura 6.12: Densidad de llegada de paquetes para el tráfico SOCCER1500 en la red 2 para un retraso de 0.05s. (Los valores de retraso en el gráfico están en **ms**). Se representan tres niveles de carga (0%: sin carga, 50% media carga y 100%: carga total). La última gráfica es una porción ampliada de la carga total entre 5000 y 35000**ms**.

### 6.5.2.- Resultados para la red 1

La tabla 6.5 muestra los resultados de la simulación de los retrasos máximos para los tráficos en la red 1. Se puede observar que sólo para el tráfico SOCCER1500 se sobrepasa el límite  $d_f$ . Esta condición sólo ocurre cuando la red está totalmente cargada (por ejemplo, 0.52% TFF para 0.1s). En general, no existen muchas diferencias en el retraso de los paquetes con los distintos retrasos máximos cuando la red no está muy cargada. Sólo con el retraso máximo de 10000 $\mu$ s el retraso experimentado por los paquetes es bastante menor que para el resto de los retrasos. Esto es debido a la utilización de la ecuación (10) del retraso que no depende de las características del tráfico. La figura 6.12 y 6.13 muestran las tasas de retraso normalizadas, es decir, el retraso máximo dividido por el retraso requerido para los tráficos LAMBS1500 y SOCCER1500. Una tasa por encima de 50% implica que el canal de reserva tiene que ser activado.

Retraso/Carga	0	25	50	75	100
<b>LAMBS1500</b>					
100000	643	1601	3191	5003	8929
50000	643	1520	3223	4671	6903
20000	643	1447	2793	4056	5816
10000	78	470	1084	1584	2166
<b>SOCCKER1500</b>					
100000	643	1574	3137	5143	(0.52%) 78470
50000	643	1479	3000	3926	24178
20000	643	1275	2348	3478	(0.05%) 10895
10000	643	1008	1956	2603	4114
<b>NEWS1500</b>					
100000	613	1420	2676	3963	7047
50000	576	1481	2997	3736	6693
20000	78	493	1048	1671	2136
10000	78	422	1015	1580	2095

Tabla 6.5: Retraso máximo de paquete en la red 1. Todos los valores están  $\mu$ s. Entre paréntesis está el valor de TFF cuando se supera  $d_{total}/2$ .

### 6.5.3.- Resultados para la red 2

En la tabla 6.6 se muestran los resultados de los retrasos máximos para los tráficos en la red 2. Los resultados son muy similares a la red 1.

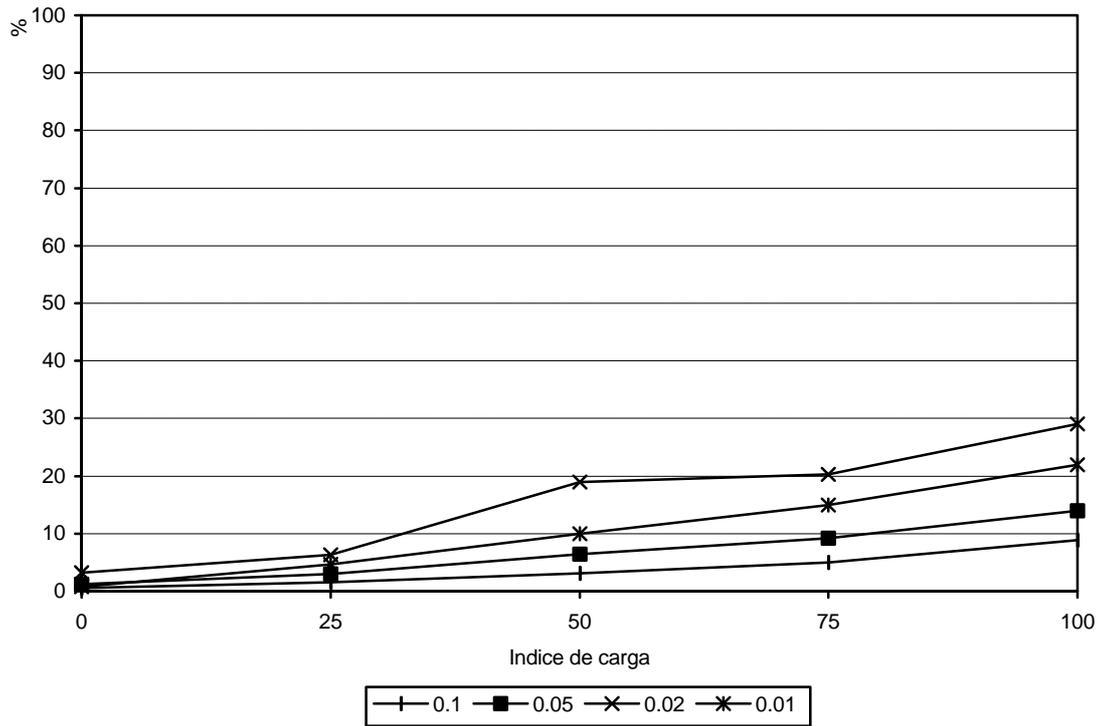


Figura 6.12: Tasas de retraso normalizadas para el tráfico LAMBS1500 en la red1.

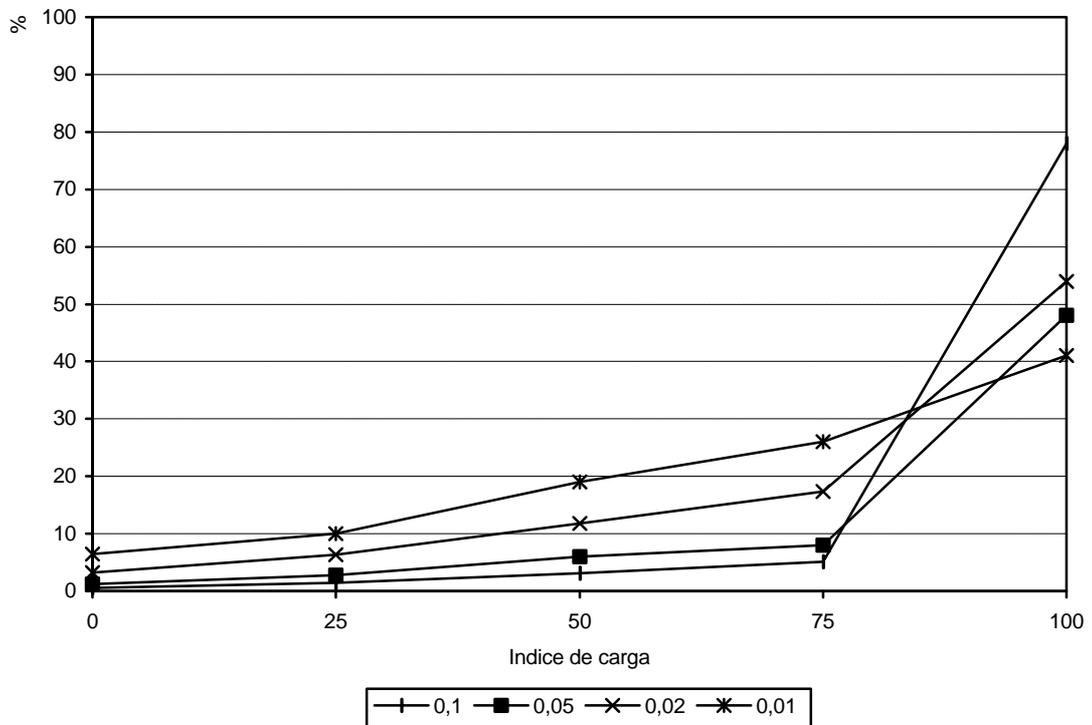


Figura 6.13: Tasas de retraso normalizadas para el tráfico SOCCER1500 en la red1.

Retraso/Carga	0	25	50	75	100
<b>LAMBS1500</b>					
400000	1537	3225	6910	12835	36390
200000	1537	3340	6770	12638	26687
100000	1537	2932	7396	11275	19851
50000	1537	2868	5932	11652	16728
<b>SOC CER1500</b>					
400000	1537	3112	7388	11051	42521
200000	1537	2852	7689	11556	23521
100000	1537	2955	7428	10636	(0.16%) 57221
50000	1537	2638	6912	9010	(0.03%) 25661
<b>NEWS1500</b>					
400000	1487	2789	6548	10632	20779
200000	1464	3063	7021	9849	19410
100000	1260	2993	6301	10103	17377
50000	187	1037	2636	4534	6620

Tabla 6.6: Retraso máximo de paquete en la red 2. Todos los valores están **ms**. Entre paréntesis está el valor de TFF cuando se supera  $d_{total}/2$ .

### 6.5.4.- Resultados para la red 3

Finalmente, en la tabla 6.7 están los resultados de simulación para la red 3. Los retrasos son muy pequeños debido al elevado ancho de banda y pequeño (y fijo) tamaño del paquete. En esta red, semejante a una ATM, los retrasos obtenidos están muy cercanos a los retrasos máximos cuando la red está totalmente cargada. La razón es que el pequeño tamaño del paquete hace que el tráfico sea mucho más fluido con lo que la red se comporta más parecida a un flujo y las ecuaciones de retraso dan mejores estimaciones. Para el tráfico SOCCER1500 la tasa de falsos fallos es alta.

Retraso/Carga	0	25	50	75	100
<b>LAMBS1500</b>					
100000	15.27	93.06	101.85	103.74	47859
50000	15.27	99.25	101.86	103.04	(0.42%) 26735
10000	15.27	92.33	96.41	103.20	3297
5000	15.27	93.54	99.54	100.53	373
<b>SOC CER1500</b>					
100000	15.27	70.11	104.6	105.46	(21.05%) 96091
50000	15.27	71.21	104.17	104.76	(13.71%) 46569
10000	15.27	70.43	102.72	103.35	(0.36%) 8201
5000	15.27	69.96	103.48	104.53	(0.87%) 3392
<b>NEWS1500</b>					
100000	15.27	98.33	98.19	100.05	5424
50000	15.27	82.20	103.55	102.66	1730
10000	15.27	97.24	100.12	96.95	307.58
5000	15.27	92.35	94.15	99.59	173.2

Tabla 6.7: Retraso máximo de paquete en la red 3. Todos los valores están **ms**. Entre paréntesis está el valor de TFF cuando se supera  $d_{total}/2$ .

### 6.5.5.- Conclusiones

En general, los valores de los retrasos de los paquetes en las simulaciones son bastante similares a los del artículo de Clark [Clark92]. La primera cuestión que surge es por qué los retrasos de los paquetes son mucho menores que sus retrasos máximos. La principal razón se debe a la alta variabilidad de la tasa del tráfico. Otra razón es la caracterización del tráfico (el modelo del flujo) que es una aproximación muy basta de la dinámica del tráfico y que las ecuaciones de retraso son muy pesimistas.

Las simulaciones también muestran que no es necesario inyectar paquetes “estoy vivo” usando tráfico MPEG con lo que no se sobrecarga las líneas y se simplifica la implementación del protocolo.

Con estos resultados experimentales la reserva de recursos en el canal de reserva se puede reducir más disminuyendo el umbral de detección. Por ejemplo, en la red 1 para el tráfico NEWS con un retraso de  $d_{total}=0.05s$  se puede utilizar un umbral de detección de  $d_f=0.01s$ , ya que no hay ningún paquete que sobrepase este valor (véase la tabla 6.5). Esto proporciona para el canal de reserva un retraso máximo de  $d_B=d_{total}-d_f=0.4s$ . La reserva para este retraso es 0.99Mb/s (para el retraso original de 0.025s era una reserva de 1.73Mb/s). Esto da una reserva total de  $R_{total}=R_A+R_B=1.89Mb/s$  que ahorra 1.57Mb/s sobre el esquema A|B (46% de ahorro). Para un retraso total de  $d_{total}=0.01s$  el ahorro puede ser de hasta un 60%.

### 6.5.6.- Resultados para leaky bucket

Los resultados de las simulaciones para el modelo de tráfico *leaky bucket* están muy cercanos a sus retrasos máximos, como se puede ver en la tabla 6.8. El índice de carga tiene poca influencia en el retraso máximo. La principal razón es que modelo *leaky bucket* limita el tráfico en la fuente, de tal forma que los paquetes son inyectados en la red con una tasa máxima de  $r$  b/s. En cambio, el modelo *token bucket* permite inyectar el tráfico con una tasa pico  $p$  en un intervalo de tiempo determinado. Esto implica que los paquetes son almacenados y retrasados en la fuente. Para retrasos pequeños, por ejemplo en el tráfico NEWS1500, los resultados son los mismos que para el flujo *token bucket* porque la reserva de ancho de banda es mayor que el parámetro  $r$ , con lo que no se encola ningún paquete. En general, estos resultados muestran que con el modelo *leaky bucket* este esquema es menos eficiente, y en algunos casos la TFF es demasiado alto para que sea útil (por ejemplo en el tráfico SOCCER1500). Los resultados parecen indicar que para tráficos con poca variabilidad (LAMBS1500 y NEWS1500) puede ser viable su uso. En la red 3 el retraso máximo de un paquete (véase tabla 6.10) está bastante cercano a su retraso

máximo. Esto demuestra que este nuevo esquema no es útil en redes ATM con un modelo de tráfico *leaky bucket*.

Retraso/ Carga	0	25	50	75	100
<b>LAMBS1500</b>					
<b>100000</b>	(2.3%) 61293	(2.3%) 61428	(2.5%) 61887	(2.5%) 62142	(3.1%) 63266
<b>50000</b>	24607	24704	24960	(0.14%) 25811	(0.32%) 26129
<b>20000</b>	3092	3717	5416	5673	7091
<b>10000</b>	78	470	1084	1584	2166
<b>SOC CER1500</b>					
<b>100000</b>	(15.2%) 85357	(15.2%) 85496	(15.2%) 85815	(15.3%) 87260	(15.3%) 87609
<b>50000</b>	(3%) 37051	(3%) 37131	(3.2%) 37292	(3.4%) 38126	(3.9%) 40328
<b>20000</b>	9706	9790	(0.1%) 10763	(0.2%) 11420	(0.4%) 11472
<b>10000</b>	1768	1889	2899	3056	4539
<b>NEWS1500</b>					
<b>100000</b>	39587	39587	40514	41272	41463
<b>50000</b>	10644	10644	11244	11981	12722
<b>20000</b>	78	493	1048	1671	2136
<b>10000</b>	78	422	1015	1580	2095

Tabla 6.8: Retraso máximo de paquete en la red 1 usando *leaky bucket*. Todos los valores están **ms**. Entre paréntesis está el valor de TFF cuando se supera  $d_{total}/2$ .

Retraso/ Carga	0	25	50	75	100
<b>LAMBS1500</b>					
<b>400000</b>	169863	170208	170453	174827	176992
<b>200000</b>	76493	77154	77214	79836	84386
<b>100000</b>	27545	27845	29295	32740	35682
<b>50000</b>	3762	4769	8400	11890	17369
<b>SOC CER1500</b>					
<b>400000</b>	(19.6%)345751	(19.6%)345962	(19.7%)347201	(19.7%)349473	(20.1%)356340
<b>200000</b>	(15.2%)155496	(15.3%)156086	(15.3%)157168	(15.5%)158966	(15.6%)165478
<b>100000</b>	(3.1%)61242	(3.2%)62548	(3.7%)64178	(4.7%)67097	(6.6%)72142
<b>50000</b>	17448	18128	20282	24953	(0.1%)27607
<b>NEWS1500</b>					
<b>400000</b>	105472	106754	107457	111030	115530
<b>200000</b>	38468	38619	39946	41853	47206
<b>100000</b>	5130	6359	8849	12843	18322
<b>50000</b>	187	1037	2636	4534	6620

Tabla 6.9: Retraso máximo de paquete en la red 2 usando *leaky bucket*. Todos los valores están **ms**. Entre paréntesis está el valor de TFF cuando se supera  $d_{total}/2$ .

Retraso/ Carga	0	25	50	75	100
<b>LAMBS1500</b>					
<b>10000</b>	(23.4%) 94846	(23.4%) 94861	(23.5%) 94847	(23.5%) 94897	(23.5%) 94923
<b>50000</b>	(18.3%) 47145	(18.3%) 47145	(18.3%) 47173	(18.3%) 47170	(18.3%) 47194
<b>10000</b>	(10%) 8984	(10.1%) 8988	(10.2%) 9007	(10.2%) 8999	(10.3%) 9012
<b>5000</b>	(4.8%) 4213	(4.9%) 4213	(5%) 4213	(5%) 4276	(5%) 4261
<b>SOCCER1500</b>					
<b>100000</b>	(21.5%) 99198	(21.5%) 99204	(21.5%) 99233	(21.5%) 99225	(21.5%) 99256
<b>50000</b>	(16.8%) 49344	(16.8%) 49356	(16.8%) 49355	(16.8%) 49382	(16.8%) 49435
<b>10000</b>	(6%) 9625	(6%) 9625	(6%) 9626	(6%) 9677	(6%) 9659
<b>5000</b>	(4.2%) 4667	(4.2%) 4674	(4.2%) 4686	(4.2%) 4723	(4.3%) 4702
<b>NEWS1500</b>					
<b>100000</b>	(27.8%) 92695	(27.8%) 92695	(27.8%) 92712	(27.8%) 92729	(27.8%) 92743
<b>50000</b>	(25%) 45198	(25%) 45198	(25%) 45206	(25%) 45263	(25.1%) 45232
<b>10000</b>	(13.3%) 7200	(13.5%) 7212	(13.3%) 7200	(13.7%) 7204	(13.7%) 7263
<b>5000</b>	2450	2454	2486	2461	2492

Tabla 6.10: Retraso máximo de paquete en la red 3 usando leaky bucket. Todos los valores están **ms**. Entre paréntesis está el valor de TFF cuando se supera  $d_{total}/2$ .

## 6.6.- Implementación

Este esquema puede ser implementado como parte de un protocolo de transporte sobre cualquier protocolo de red, como se muestra en la figura 6.14. El FSM (*Failure Suspect Module*) se encarga de proporcionar los mecanismos para:

- calcular los retrasos exigidos al canal primario y secundario ( $d_A$  y  $d_B$ ) a partir del retraso total ( $d_{total}$ ),
- activar el canal de reserva cuando un paquete se retrasa más que  $d_f$ , y
- inyectar paquetes "estoy vivo" en la red cuando no hay tráfico.

Sin embargo, las simulaciones han demostrado que para tráfico MPEG esta inyección es innecesaria por la propia codificación, en la que nunca hay intervalos sin tráfico para enviar.

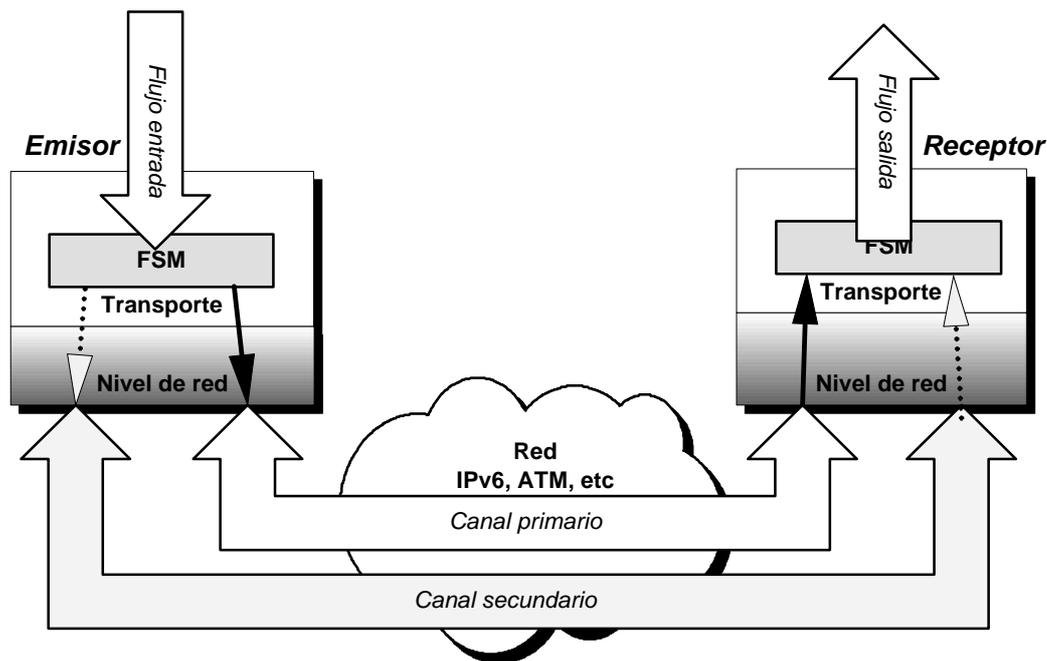


Figura 6.14: Implementación del módulo de detección de fallo FSM.

El aspecto más importante de este esquema es la detección de los fallos. Como se ha visto, se considera que hay fallo en el canal cuando un paquete no ha llegado antes de su límite  $d_f$ , considerando este valor como retraso emisor-receptor. Pero, ¿cómo puede saber el receptor si un paquete supera este retraso?. Para ello, el emisor incluye como parte de un paquete el tiempo de salida del siguiente paquete ( $t_{prox}$ ). Este valor es conocido por el emisor porque conoce el tráfico a transmitir y sabe cuándo va a inyectar el próximo paquete. Cuando este paquete llega al receptor, la condición de fallo será, por tanto, si el próximo paquete llega en un tiempo mayor a  $t_{prox} + d_f$ . Para que este mecanismo funcione correctamente es crítica una perfecta sincronización temporal entre el emisor y receptor, es decir, que utilicen una misma base de tiempos.

En la figura 6.15 se muestra un ejemplo de flujo de mensajes. Cuando se envía el primer mensaje en el tiempo  $t_1$  se indica que el próximo paquete se enviará en tiempo  $t_2$ . Cuando este paquete llega al receptor, éste sabe que el siguiente paquete llegará antes de  $t_2 + d_f$ . Si el mensaje 3 se pierde por una caída de la red el receptor sabe que este paquete tenía que llegar antes de  $t_3 + d_f$ . Cuando vence este tiempo el FSM asume que hay un fallo con lo que comienza el mecanismo de activación del canal de reserva.

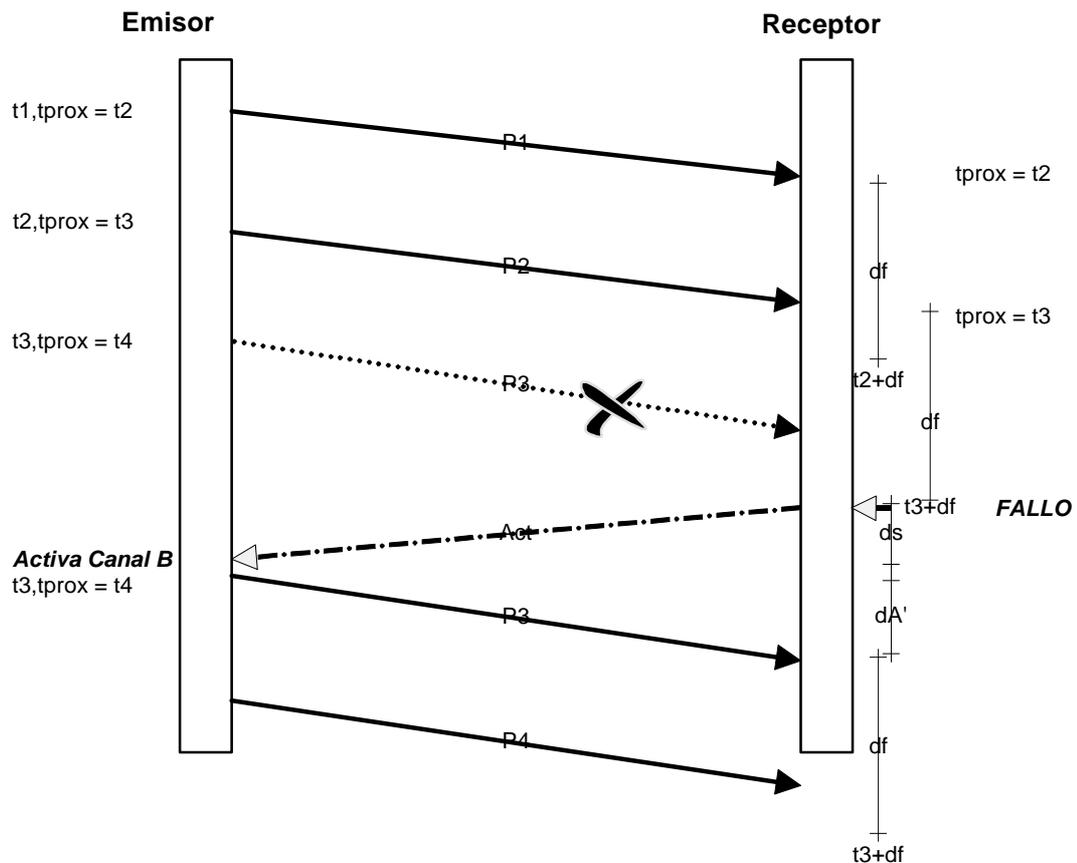


Figura 6.15: Flujo de mensajes entre emisor y receptor.

La activación del canal de reserva se producirá cuando este mensaje de activación llegue al emisor por medio de este mismo canal de reserva (véase figura 6.16). Cuando el emisor recibe este mensaje de activación del canal de reserva, empieza a transmitir a través de este canal. No se puede olvidar que el objetivo es que el retraso de este paquete entre emisor y receptor sea siempre menor que  $d_{total}$ . Como se ha comentado al principio del capítulo, no se ha tenido en cuenta el tiempo de activación del canal  $d_s$  que es el tiempo necesario para que este mensaje de activación llegue al emisor. Aunque en las simulaciones se ve que este valor es muy pequeño (varía entre  $15\text{ms}$  para la red 3 hasta  $187\text{ms}$  para la red 2) se puede obviar este valor con el siguiente razonamiento.

En el momento en que se activa el canal de reserva se tiene que el tiempo consumido es  $d_f + d_s$ . Al canal de reserva se le ha exigido un retraso máximo  $d_B$ , con lo que en teoría el retraso máximo posible será  $d_f + d_s + d_B$ . Pero cuando el emisor empieza a transmitir por el canal de reserva el primer mensaje, las colas de este canal están vacías. Esto implica que no puede haber retraso de cola, con lo que el retraso de este primer mensaje vendrá dado por la segunda ecuación del retraso del capítulo 2:

$$d'_B = Q_{delay} = \frac{(M + C_{tot})}{R} + D_{tot} \quad (\text{caso } R \geq p \geq r)$$

El valor de este retraso sólo depende de los parámetros de la red y la reserva realizada, que es bastante menor que el retraso calculado  $d_B$ . Por lo tanto, este retraso compensa el valor  $d_s$ . Para ser exacto, la suma de  $d'_B + d_s$  debe ser siempre menor a  $d_B$ . Cuando ya se ha enviado el primer mensaje, el funcionamiento será igual que para el canal primario y el siguiente mensaje estará acotado por  $d_f$ .

Por ejemplo, en la red 1 para el tráfico LAMBS1500 y un retraso total de 0.1s se tiene que el retraso exigido al canal B  $d_B$  es de 0.05s y la reserva 1.36Mbps. Con estos valores el retraso  $d'_B$  es de 0.031s, lo cual deja un margen de 0.029s para  $d_s$ . Para un retraso más exigente de 0.02s el valor  $d_B$  es 0.01s y la reserva 4.64Mbps en el tráfico NEWS1500. El valor del retraso  $d'_B$  es 0.099s lo que da un margen de 0.001s. En general los valores permitidos para  $d_s$  son generalmente suficientemente altos, aunque el valor se reduce para retrasos muy exigentes.

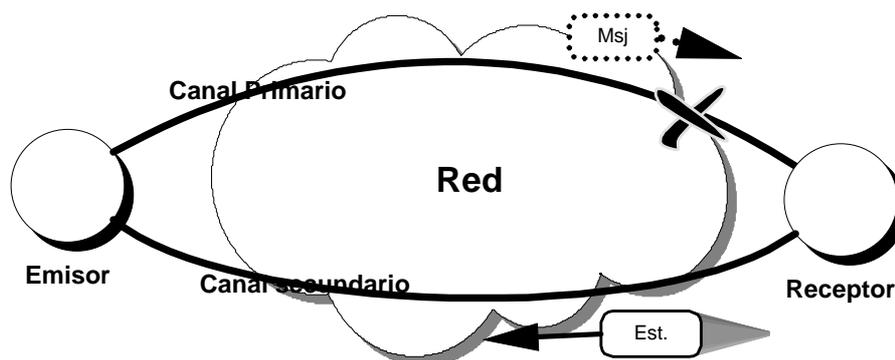


Figura 6.16: Activación del canal secundario.

La recuperación del canal primario, o la liberación de sus recursos están fuera del ámbito de esta tesis. Como se ve, la red o el nivel de transporte deben proporcionar los siguientes servicios:

- creación de canales disjuntos,
- un mecanismo de reserva que asegure retraso (como ATM, RSVP, etc),
- un mecanismo de transmisión robusto y acotado para transmitir los mensajes de activación del canal de reserva, y
- sincronización temporal entre emisor y receptor.

## 6.7.- Conclusiones

En este capítulo se ha introducido un nuevo esquema para la detección de fallo en canales de tiempo real de alta disponibilidad que proporcionan una reserva eficiente de recursos. La idea detrás de este esquema es que los límites de retraso calculados para redes en tiempo real son muy pesimistas. Normalmente, los paquetes llegan mucho antes que su retraso, tal como se ha visto. Esta característica puede ser usada para reducir las reservas en la red, asegurando un servicio de alta disponibilidad.

Este esquema se ha comparado con la reserva de recursos de las otras dos alternativas usando tráfico MPEG. De estos resultados se puede concluir que el esquema propuesto es un compromiso entre el esquema de copia múltiple y de canal de reserva con activación en fallo: se reserva más recursos que en el de copia múltiple y menos que en el de activación en fallo, pero con la penalización de que el canal de reserva se activa innecesariamente cuando ocurren falsos fallos. Se han realizado simulaciones usando los modelos de tráfico *token bucket* y *leaky bucket* con planificadores WFQ. Estas simulaciones muestran que para el modelo *token bucket* la activación del canal de reserva es muy rara: en la mayoría de las simulaciones esta tasa es 0 y siempre está por debajo del 5%. Sólo ocurren falsos fallos cuando la red está totalmente cargada (una condición muy improbable) y con retrasos muy pequeños.

El ahorro total en la red varía entre el 10% y el 35% sobre el esquema de canales de reserva. Considerando que el nodo puede usar los recursos del canal de reserva para otro tráfico no prioritario cuando el canal no está en uso, el ahorro varía entre el 10% y el 55%. El hecho de que las simulaciones muestren que los paquetes llegan mucho antes que sus retrasos máximos puede ser usado para incrementar el límite de detección de fallo  $d_f$ . Esto permite mayores valores de retraso máximo para el canal de reserva y mayores ahorros de recursos: hasta el 60%.

Para el modelo *leaky bucket* los retrasos de los paquetes están más cercanos al retraso calculado, con lo que el esquema propuesto es menos eficiente debido a la alta tasa de activación del canal de reserva.

Al final del capítulo se describe como implementar el mecanismo de detección de fallo encima de otros protocolos.



# 7.

## Conclusiones

### ***7.1.- Conclusiones***

En esta tesis, se ha abordado un aspecto crucial de las redes de tiempo real que es la optimización de la reserva de recursos para garantizar una determinada calidad de servicio. Para ello se han planteado distintas optimizaciones y mejoras, y su ámbito de aplicación.

En general, se ha partido de un tráfico conocido, es decir, que se puede estudiar. Además, las optimizaciones se han limitado siempre a un servicio garantizado.

#### ***7.1.1.- Estado de las soluciones propuestas***

En los tres primeros capítulos de esta tesis se ha introducido la problemática asociada a la transmisión en tiempo real. Se comprueba que para obtener un servicio garantizado se requiere una elevada reserva de recursos que normalmente, son infrautilizados. Esto confirma el interés de la tesis en la búsqueda de nuevos mecanismos para optimizar la reserva de estos recursos.

La aportación más significativa de esta primera parte es la introducción de un esquema que permite la comparación de los modelos de tráfico basado en las funciones de tráfico limitado. Esto permite obtener a partir de un tráfico determinado, los parámetros que especifican el tráfico de varios tipos de redes como: ATM, *Tenet Suite* o IETF/RSVP. De esta forma, se pueden ajustar los parámetros del modelo de tráfico a la carga a transmitir para que no estén sobreestimados con el consiguiente ahorro de recursos.

### 7.1.2.- Optimización de la reserva

En el cuarto y quinto capítulo se introducen mecanismos que permiten optimizar la reserva de ancho de banda de un enlace y la memoria de los nodos. Esto es posible introduciendo el concepto de los puntos envolventes.

A partir de un tráfico conocido y utilizando la envolvente empírica se obtiene una caracterización del tráfico que se ha denominado puntos envolventes. Estos puntos han permitido desarrollar nuevos mecanismos para la optimización de los recursos. El éxito de este enfoque es que el número de puntos envolventes para los tráficos MPEG estudiados es bajo (menor que cien), lo que permite:

- La optimización del ancho de banda *on-line* con un algoritmo de coste computacional acotado y muy rápido:  $O(\log m)$ , donde  $m$  es el número de puntos. Esto implica que en cinco o seis iteraciones se puede encontrar la reserva óptima conociendo los parámetros de la red. Esta reserva óptima coincide con los métodos de optimización utilizados hasta ahora, como por ejemplo, el introducido por Reibman et al. [Reibman95], que eran iterativos y no acotados, y que se basaban en recorrer todo el vídeo. Esto hacía imposible su utilización *on-line*, como por ejemplo en las aplicaciones de vídeo bajo demanda.
- Estos puntos se pueden transmitir a los nodos con lo que éstos disponen de información sobre las características de los canales que tienen. Con esta información y utilizando el algoritmo desarrollado en el capítulo 5, se puede optimizar la reserva de *buffer* en los nodos que mejora ampliamente el establecido por Presti et al. [Presti99], que usa un modelo de tráfico ON-OFF. Las comparaciones realizadas usando tráficos MPEG demuestran que la ganancia es muy alta y puede llegar hasta el 95%.
- Además, se proponen dos soluciones para la aplicación de la optimización anterior a redes con varios nodos. La primera se basa en optimizar la reserva en una subred, agrupando los canales a la entrada en un macrocanal con un ancho de banda determinado. El tráfico a la entrada de la subred necesita pasar por un conformador *leaky bucket* multinivel. Este esquema permite balancear de forma dinámica la optimización entre ancho de banda o memoria en función de la carga de cada subred. En la segunda solución, se establece una cota a la distorsión del tráfico, lo que implica un conjunto de puntos envolventes equivalentes al original. Con estos puntos se puede aplicar el método de optimización visto.

Otro aspecto clave de estas optimizaciones es que no plantean nuevos planificadores ni modelos de tráfico lo cual simplifica su utilización e implantación.

Permite su uso con modelos de tráfico *leaky bucket* o *token bucket* y planificadores WFQ o similares, como los propuestos por IETF/RSVP.

### **7.1.3.- Optimización con canales de reserva**

La última aportación de esta tesis es un nuevo mecanismo de activación de canales de reserva para la optimización de la reserva a realizar. Si ya la reserva de recursos es alta con un solo canal, si se tiene que proporcionar alta disponibilidad, la reserva de recursos será mayor al tener que crear nuevos canales de reserva.

Este nuevo esquema se basa en la idea de que las funciones para obtener el retraso máximo son muy pesimistas. Es normal que los paquetes lleguen mucho antes, con lo que sólo una pequeña porción de los paquetes están cercanos al retraso máximo.

En consecuencia, con el objetivo de que la reserva conjunta sea menor, se ha desarrollado un nuevo mecanismo de activación de los canales de reserva que se basa en sospechar el fallo antes de que realmente ocurra. Se ha denominado a este mecanismo activación anticipada del canal de reserva. Esto permite que los retrasos del canal primario sean mayores con lo que implica una menor reserva. Las pruebas realizadas con simulación demuestran que este esquema implica un ahorro de recursos entre el 20 y el 60 % sobre los otros esquemas comparados, sin perder la garantía de servicio.

### **7.1.4.- Simulación y carga**

Con el objetivo de probar todos los esquemas introducidos en esta tesis se ha desarrollado un simulador denominado RTNetSim. Este simulador incluye:

- la implementación de los algoritmos de optimización descritos en los capítulos cuarto y quinto, y
- ha permitido simular las redes detalladas en el capítulo sexto para probar el nuevo esquema de detección de fallos.

Además, la implementación de este simulador ha servido para comprobar que los algoritmos descritos en la tesis funcionan correctamente.

Para las pruebas realizadas se han utilizado las trazas de tráfico MPEG recopiladas por O. Rose, lo que ha permitido que las pruebas sean más realistas al utilizar tráfico reales.

## 7.2.- Líneas de trabajo futuras

Como se comenta a lo largo de la tesis son muchas las líneas de trabajo que quedan abiertas o por profundizar a partir de los resultados. El punto más interesante es la aplicación de los puntos envolventes a otros ámbitos. En concreto, los temas que parecen más interesantes para futuras líneas de trabajo son:

- Aplicar el modelo de puntos envolventes con un servicio no garantizado (estadístico). Dado que el servicio garantizado requiere la reserva de muchos recursos, muchas aplicaciones no críticas pueden tolerar un servicio estadístico lo que permite una reducción en la reserva de recursos. Con los puntos envolventes los nodos disponen de mayor información acerca del tráfico que transmiten con la cual poder calcular las reservas necesarias para obtener un determinado nivel de servicio estadístico.
- Ver las propiedades de los puntos envolventes como modelo o caracterización del tráfico VBR. Realizar, por tanto, un estudio estadístico de los puntos envolventes, caracterización de las imágenes, etc.
- Todos los esquemas presentados requieren el conocimiento previo del tráfico a transmitir con lo que no puede ser usado para la transmisión de imágenes en directo. Siguiendo el estudio del punto anterior, ver si es posible establecer categorías en las transmisiones y con ello poder definir el tráfico con una serie de puntos envolventes con los que optimizar las reservas. Por ejemplo, sería posible distinguir entre una transmisión de un debate televisivo o una retransmisión de un partido de tenis y cada una tendría un patrón de puntos envolventes predeterminados.
- Utilización de los puntos envolventes para la optimización de la reserva y configuración de los servidores de medios para vídeo bajo demanda. Con el objeto de garantizar una transmisión sin interrupciones, se tiene que garantizar en el servidor un ancho de banda del disco y memoria intermedia en el procesador, aparte los recursos de la red [Makaroff97][Chang94]. Para que un servidor de medios admita una nueva petición, existe un control de admisión que comprueba si existen los recursos necesarios para poder transmitir el medio. Se está por tanto, con otro problema de optimización de recursos, con la ventaja de que están localizados en un único servidor.
- Tal como se comenta en el capítulo 5, ver si se puede establecer analíticamente como varían los puntos envolventes a lo largo de la red.

En el tema de los canales de reserva se pueden introducir ciertas mejoras:

- Profundizar en la idea de que cuando se empieza a transmitir en el canal de reserva éste no tiene tráfico con lo que el retraso del primer paquete es mínimo. Con este planteamiento se podría disminuir la reserva en el canal secundario.

En la línea práctica, sería interesante implementar todos estos esquemas en una red en tiempo real usando las distintas implementaciones de RSVP sobre Linux:

- Comprobar en la práctica el funcionamiento del esquema de detección anticipada de fallos introducido en el capítulo 6, implementado el módulo FSM descrito al final de este capítulo.
- Implementar una red para la transmisión de vídeo baja demanda que englobe todas las soluciones presentadas en esta tesis siguiendo el modelo de la red ORNet, tal como se presenta en el Apéndice 3.



# Apéndice 1:

## Trazas MPEG usadas

### A1.1.- Introducción

A lo largo de esta tesis se utilizan las muy conocidas trazas MPEG-1 provenientes del conjunto de secuencias estudiadas por O.Rose de la Universidad de *Wurzburg* [Rose95a]. Estas secuencias han sido codificadas usando el *MPEG-encoder* (versión 1.3) de *Berkeley* con los parámetros descritos en la tabla A1.1. La codificación es de 25 tramas por segundo y utiliza codificación VBR.

<b>Encoder Input</b>	384 x 288 pel
<b>Colour format</b>	YUV (4:1:1, resolution of 8 bits)
<b>Quantization values</b>	I=10, P=14, B=18
<b>Pattern</b>	IBBPBBPBBPBB
<b>GOP Size</b>	12
<b>Motion vector search</b>	'Logarithmic' / 'Simple'
<b>Reference Frame</b>	'Original'
<b>Slices</b>	1
<b>Vector/range</b>	Half pel / 10

Tabla A1.1 : Parámetros del codificador MPEG.

La mayoría de los tráfico usados tienen 40000 tramas, que son aproximadamente media hora de vídeo. Los tráfico descritos por Rose y utilizados en la tesis están enumerados en la tabla A1.2.

### A1.2.- Codificación de vídeo MPEG

El esquema de codificación MPEG de vídeo es ampliamente usado para compresión de vídeo y audio. El *Moving Picture Experts Group* (MPEG) ha sido el grupo que ha introducido los distintos estándares para la codificación de vídeo. En principio, el algoritmo de codificación MPEG fue desarrollado para el almacenamiento de vídeo comprimido [Gall91].

<b>Películas</b>	
<i>Bond</i>	James bond: Goldfinger
<i>Dino</i>	Parque Jurásico
<i>Lambs</i>	El silencio de los corderos
<i>Starwars</i>	La guerra de las galaxias
<i>Terminator</i>	Terminator 2
<i>Movie</i>	Movie preview
<b>Deportes</b>	
<i>Atp</i>	Final de tenis ATM: Becker – Sampras
<i>Race</i>	Carrera de fórmula 1 en Hockenheim /Alemania 1994
<i>Showl</i>	Final de la <i>super bowl</i> 1995: San Diego – San Francisco
<i>Soccer</i>	Final de la copa del mundo de 1994: Brasil contra Italia.
<i>Soccer2</i>	Partido de la copa del mundo de 1994: Alemania contra Bélgica.
<b>Noticiarios / debates</b>	
<i>News</i>	Telediario alemán
<i>News2</i>	Telediario alemán
<i>Talk1</i>	Show alemán
<i>Talk2</i>	Debate político
<b>Otras secuencias de TV</b>	
<i>Simpsons</i>	Los Simpsons. Dibujos animados.
<i>Asterix</i>	Asterix. Dibujos animados.
<i>Mr.bean</i>	Tres episodios de Mr. Bean
<i>Mtv1</i>	Vídeos musicales
<i>Mtv2</i>	Vídeos musicales
<b>Cámara fija</b>	
<i>Settop</i>	Estudiante sentado frente a una estación de trabajo

Tabla A1.2: Tráficos MPEG.

Hay dos estándares: MPEG-I y MPEG-II, en el que las funcionalidades de MPEG-I son un subconjunto de las de MPEG-II, que permite codificación por capas. Las trazas usadas en esta tesis son del tipo MPEG-I.

La compresión MPEG se realiza reduciendo la redundancia espacial y temporal del vídeo original. Las redundancias espaciales son reducidas por medio de transformadas y codificación por entropía, y las temporales por medio de la predicción de las tramas futuras usando vectores de movimiento. Esto se consigue usando tres tipos de tramas:

- Tramas I: compresión de una trama individualmente, basado en la transformada discreta del coseno (DCT) y codificación por entropía.
- Tramas P: se usa un esquema de codificación similar a las tramas I, pero con la adición de la compensación por movimiento con respecto a las tramas I o P anteriores.

- Tramas B: son similares a las tramas P, excepto que la compensación puede ser respecto a las tramas I o P anteriores, las siguientes o la interpolación entre ellas.

Normalmente, las tramas I requieren más bits que las tramas P. Las tramas B son de menor tamaño. Debido a la técnica de codificación MPEG, las tramas son dispuestas en una secuencia periódica determinada, esto es, “IBBPBB” o “IBBPBBPBBPBB”, que se denomina GOP (*Group of pictures*). Este esquema de codificación conduce a una variedad de propiedades estadísticas que son propias de este tráfico como las reflejadas en la tabla A1.3 calculada por Rose [Rose95a].

Secuencia	Tasa Compr. X:1	Tramas			GOPs			Tasa		Hurst Exp.
		Media [bits]	CoV	Pico/Media	Media [bits]	CoV	Pico/Media	Media [Mbps]	Pico [Mbps]	
Asterix	119	22348	0.90	6.6	262282	0.47	4.0	558718	3684400	0.81
Atp	121	21890	0.93	8.7	262648	0.37	3.0	547246	4771400	0.88
Dino	203	13078	1.13	9.1	156928	0.40	4.0	326953	2990800	0.88
Lambs	363	7312	1.53	18.4	87634	0.60	5.3	182788	3355600	0.89
Mr. Bean	150	17647	1.17	13.0	211368	0.50	4.1	441177	5726800	0.85
Mtv	134	19780	1.08	12.7	237378	0.70	6.1	494512	6285200	0.89
News	173	15358	1.27	12.4	184299	0.47	6.0	150767	1155000	0.79
Race	86	30749	0.69	6.6	369060	0.38	3.6	768727	5060400	0.99
Settop	305	6031	1.92	7.7	72379	0.18	2.0	150775	1160967	0.53
Simpsons	143	18576	1.11	12.9	222841	0.43	3.8	464389	6009400	0.89
Soccer	106	25110	0.85	7.6	301201	0.48	3.9	678213	4679400	0.91
Starwars	130	15599	1.16	11.9	187185	0.39	5.0	232829	3120400	0.74
Talk1	183	14537	1.14	7.3	174278	0.32	2.7	363420	2669200	0.89
Talk2	148	17914	1.02	7.4	214955	0.27	3.1	447874	3318800	0.73
Terminator	243	10904	0.93	7.3	130865	0.35	3.1	272618	1989000	0.89

Tabla A1.3: Estadísticas tráfico Rose.

En general, las secuencias típicas de TV como deportes, noticias y vídeos musicales tienen una relación pico/media muy alta comparado con las películas. Esto es debido a los movimientos rápidos de muchos pequeños objetos

### A1.3.- Descripción de los tráfico

De los tráfico descritos en el trabajo de Rose a continuación se detallan los más usados en la tesis. En algunos casos se ha usado sólo una porción del tráfico para que la simulación no fuese excesivamente larga. Los tráfico seleccionados representan varios tipos de tráfico, desde un tráfico con poca variabilidad como pueda ser el caso de un noticiero, o de alta variabilidad como pueda ser un partido de fútbol. Para cada uno de estos tráfico se le ha asociado un identificador que se ha usado a lo largo de la tesis.

## SOCCKER

Este tráfico corresponde a la final de la copa del mundo de 1994: Brasil contra Italia. Es un tráfico compuesto de 40000 tramas de tasa muy variable, como se muestra en la figura A1.1a. Además, se muestran las tramas I, P y B por separado. En la figura A1.1b se muestra el tráfico acumulado y la envolvente empírica.

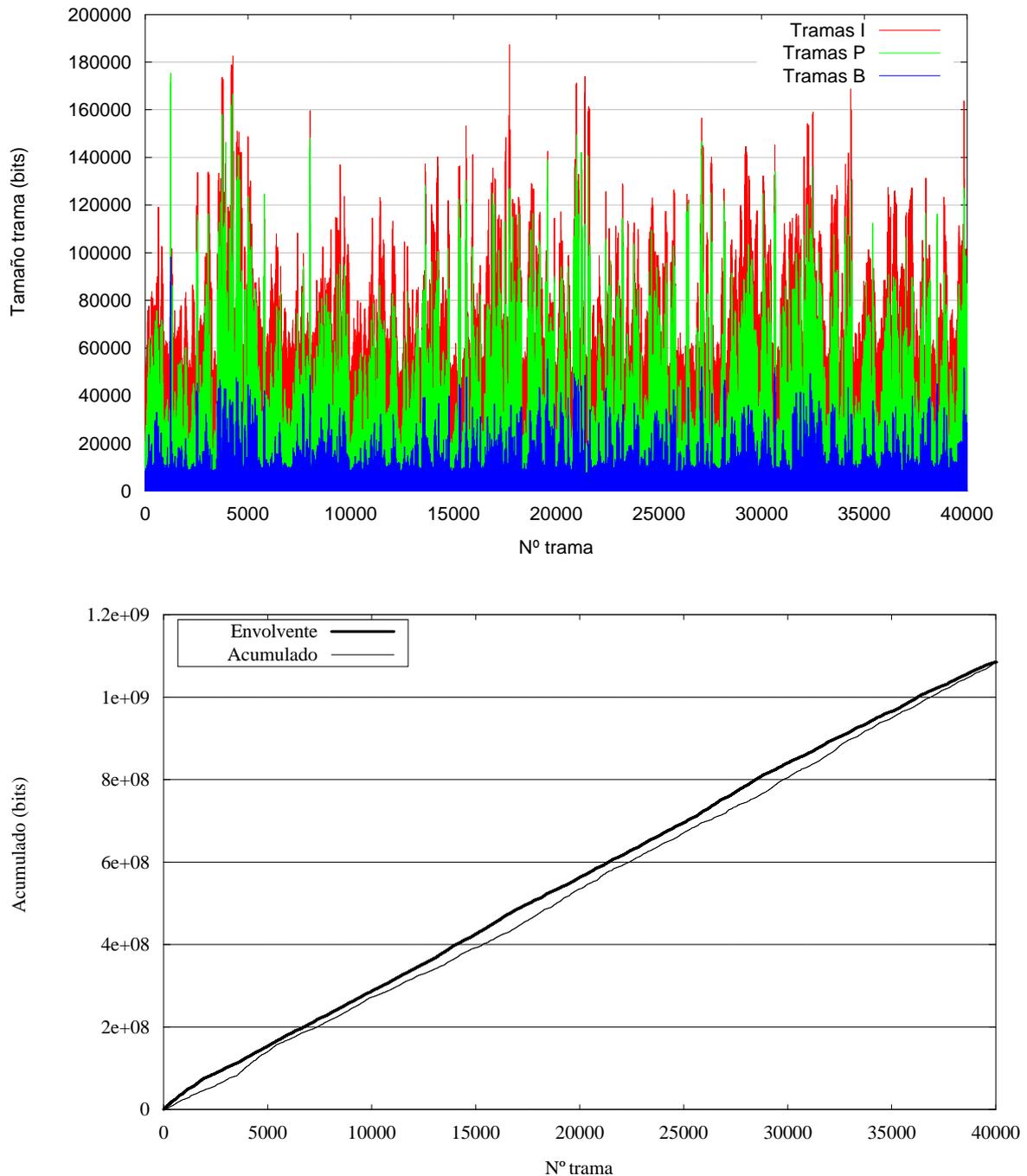


Figura A1.1: (a) Tramas del tráfico SOCCER, (b) tráfico acumulado y envolvente empírica.

## LAMBS

Este tráfico corresponde a la película “El silencio de los corderos”. Es un tráfico con tasa bastante variable con un gran pico sobre la trama 22500.

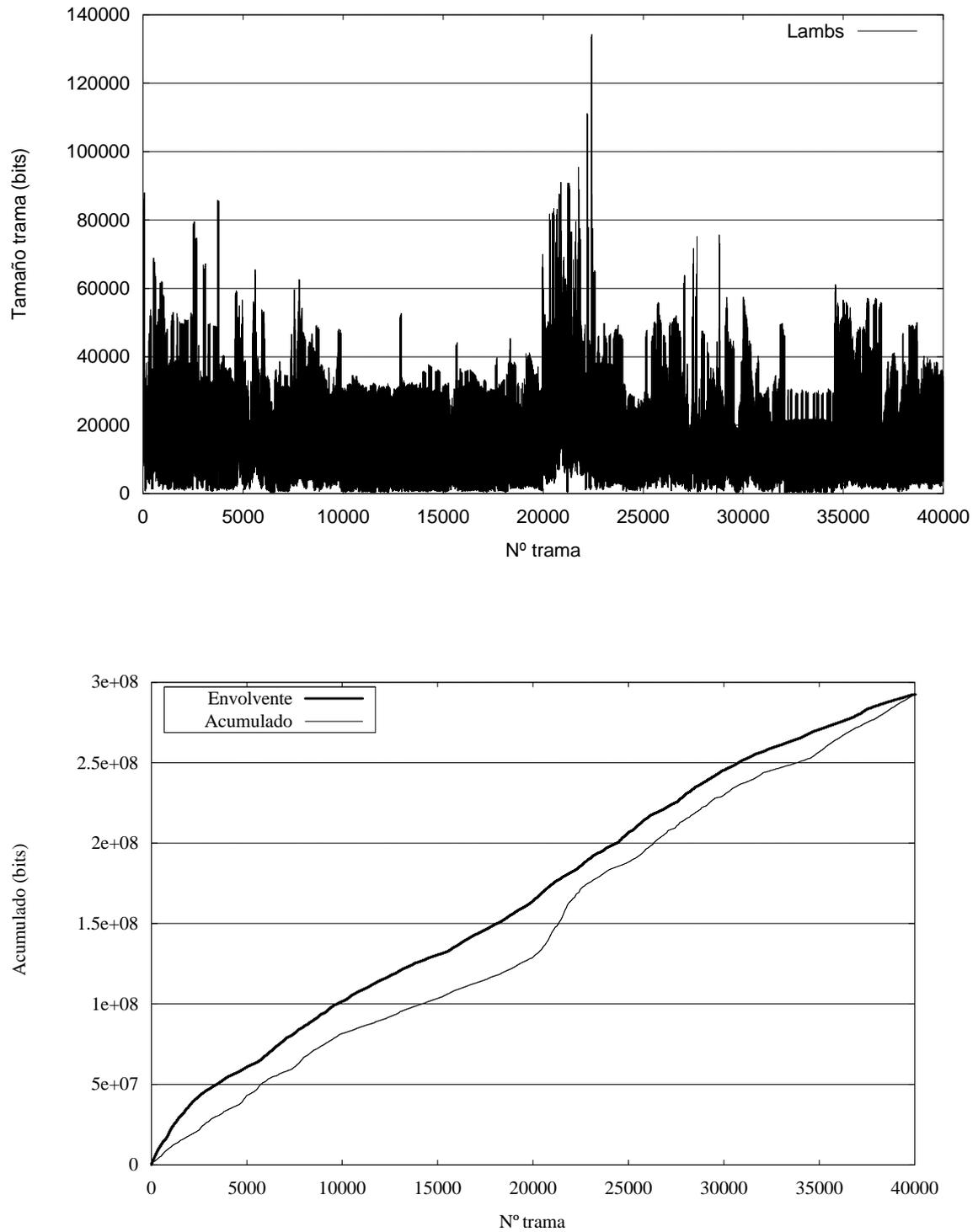


Figura A1.2: (a) Tramas del tráfico LAMBS, (b) tráfico acumulado y envolvente empírica.

## NEWS

Este tráfico corresponde a 5000 tramas de un noticiario alemán. Considerando las tramas en conjunto, es un tráfico bastante regular.

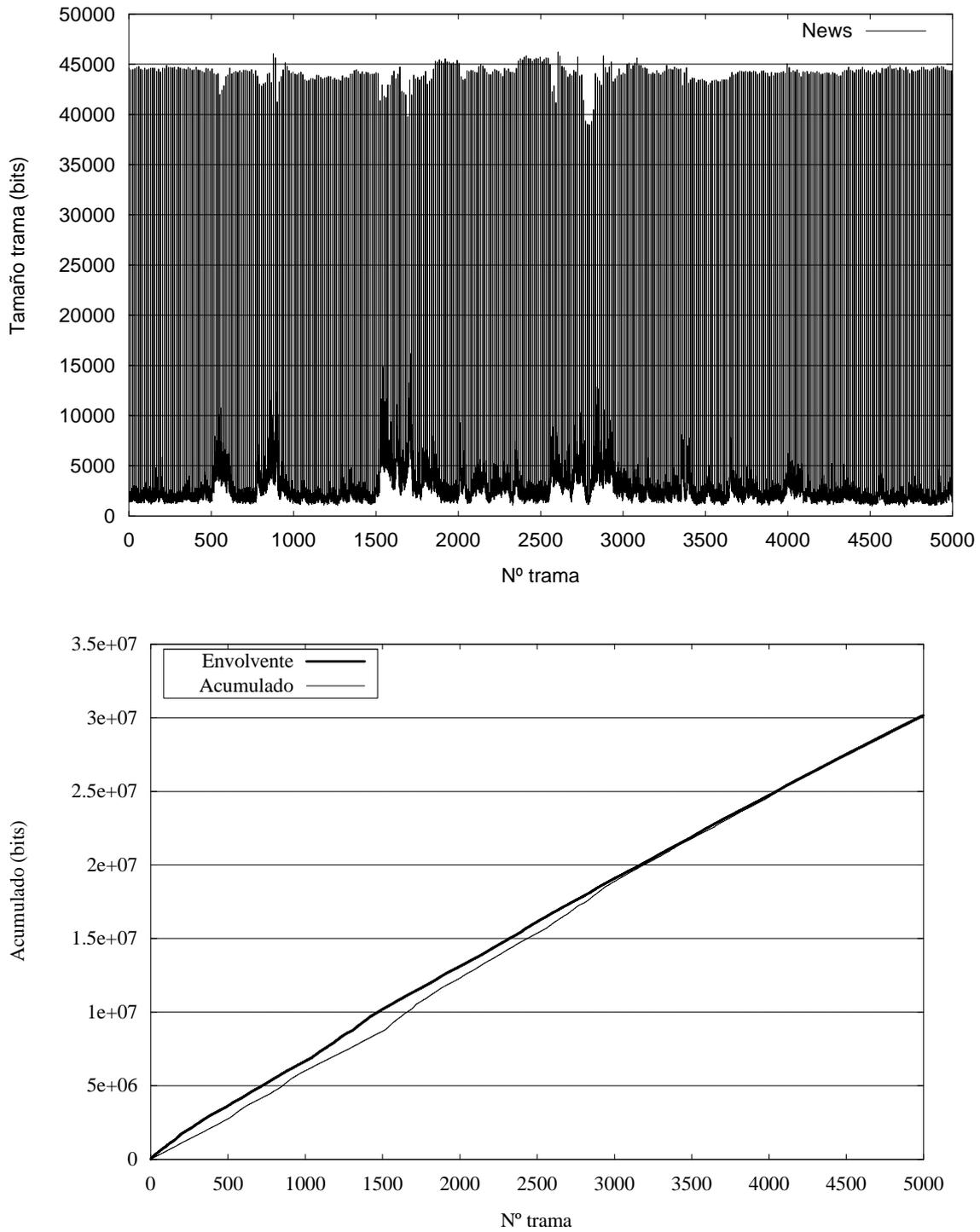


Figura A1.3: (a) Tramas del tráfico NEWS, (b) tráfico acumulado y envolvente empírica.

**ASTERIX**

Este tráfico corresponde a los dibujos animados de Asterix. Es un tráfico muy variable pero no tanto como el SOCCER.

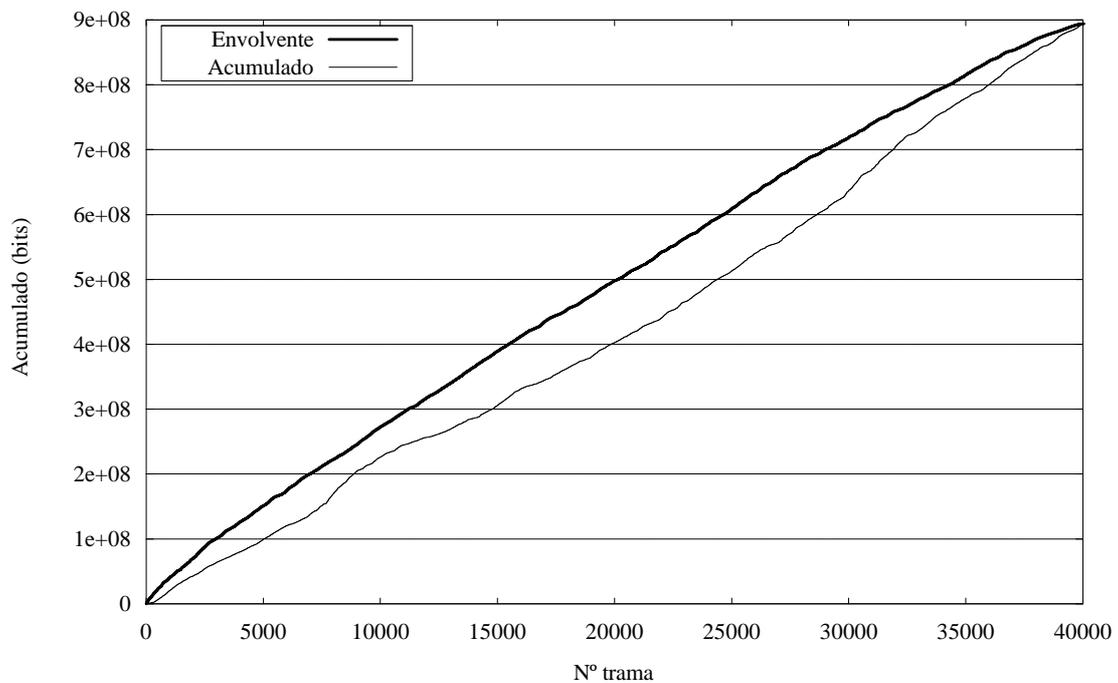
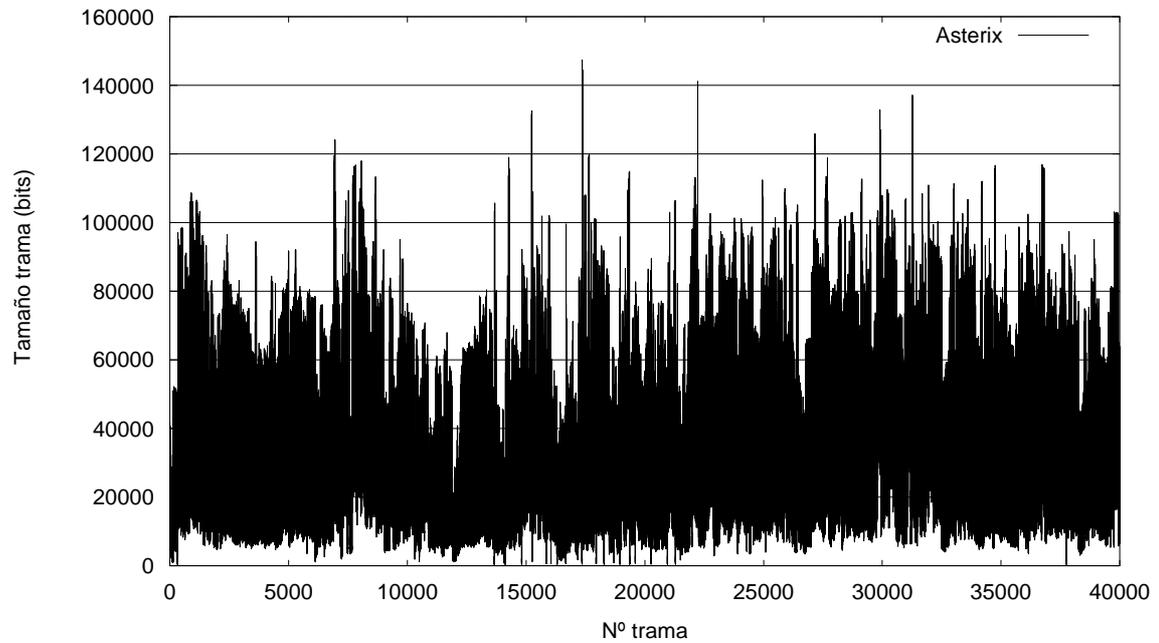


Figura A1.4: (a) Tramas del tráfico ASTERIX, (b) tráfico acumulado y envolvente empírica.

## STARWARS2

Este tráfico corresponde a la película “La guerra de las galaxias”. Este tráfico ha sido utilizado muy a menudo en la bibliografía de transmisión en tiempo real. Tiene un gran pico al principio siendo a continuación menos variable.

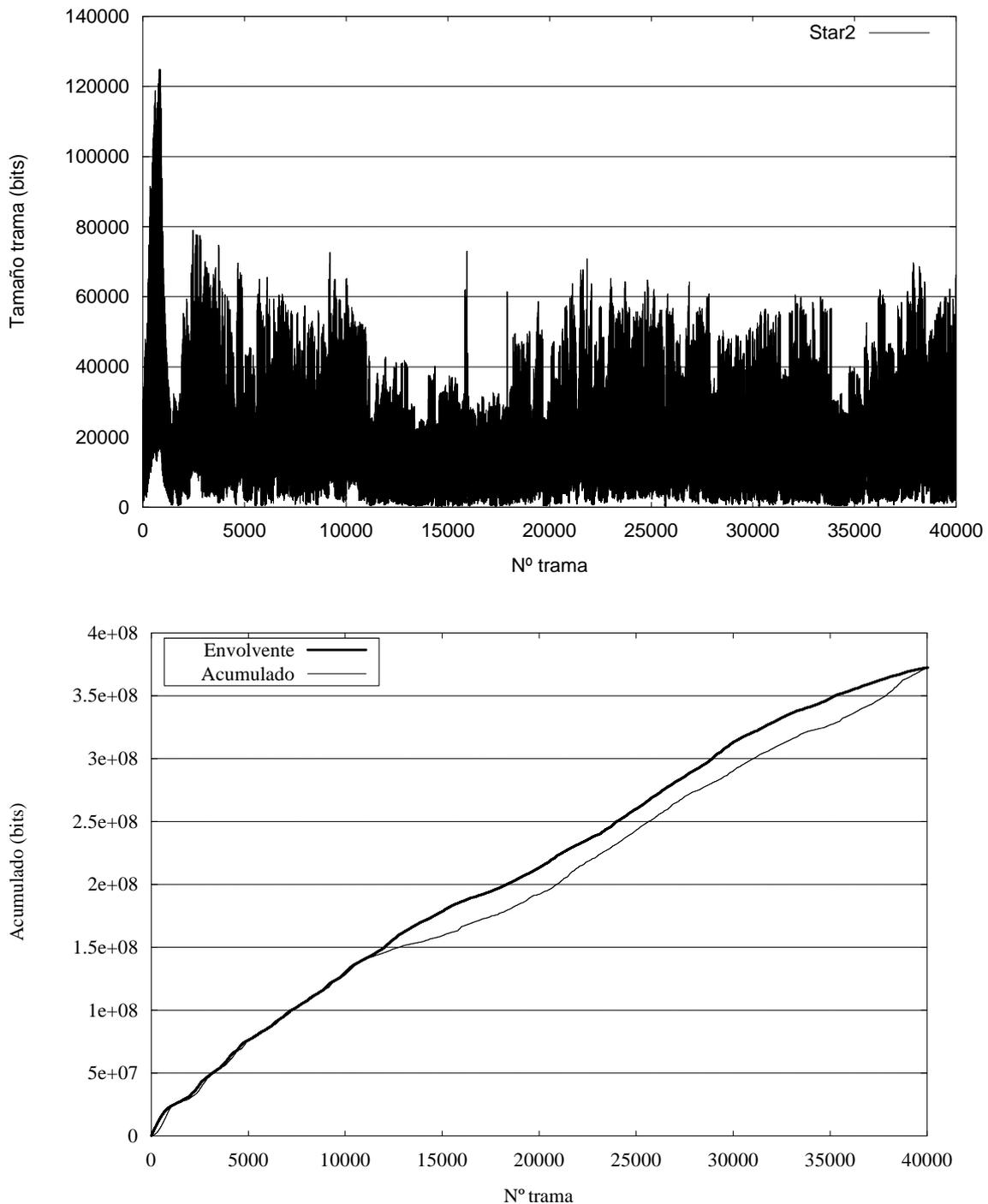


Figura A1.5: (a) Tramas del tráfico STARWARS2, (b) tráfico acumulado y envolvente empírica.

**LAMBS1500**

Este tráfico es una parte de la película “El silencio de los corderos”. En este caso, se han cogido las 1500 primeras tramas. Es un tráfico con alta variación de la tasa como se puede ver en la figura A1.6b.

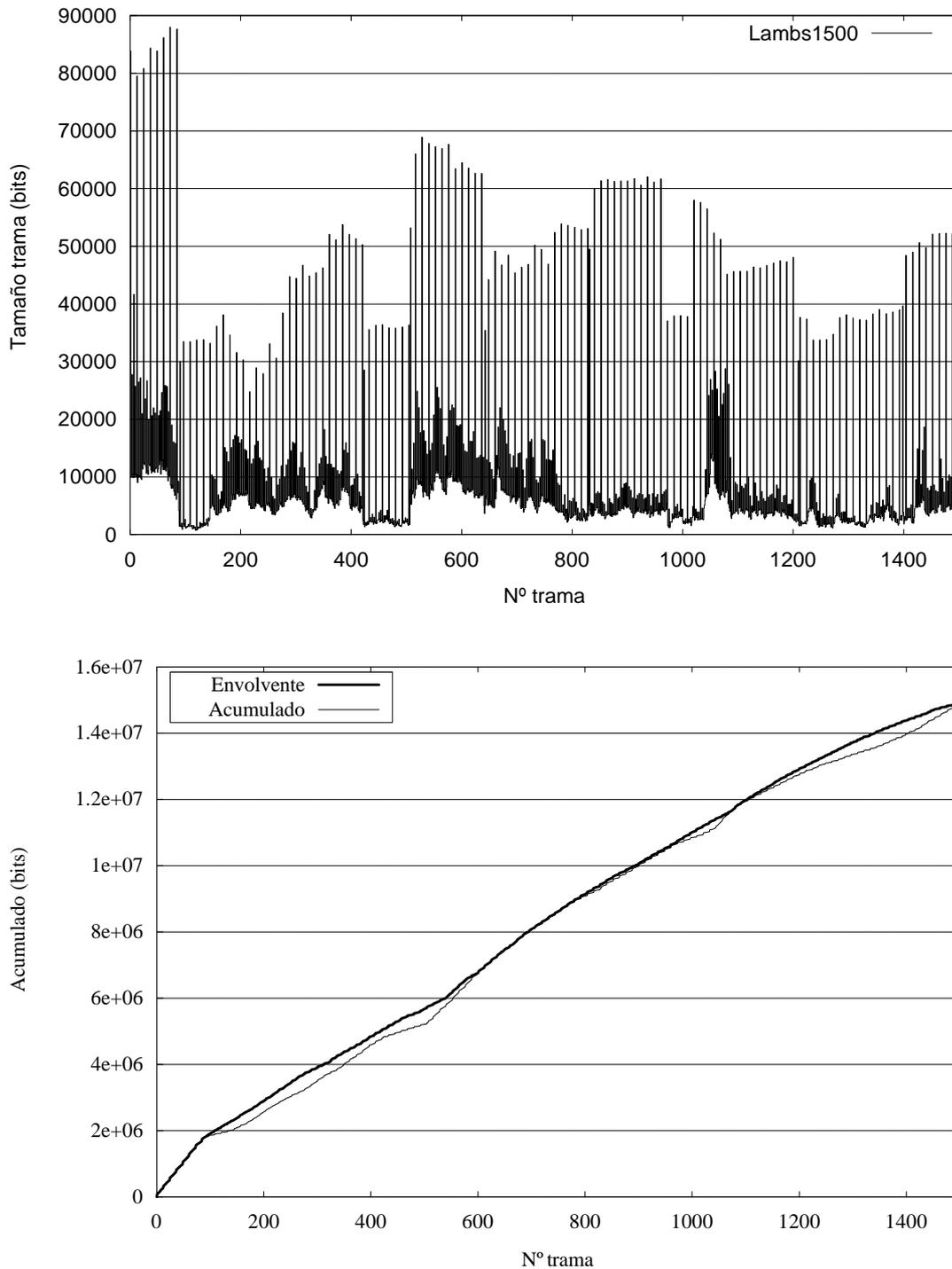


Figura A1.6: (a) Tramas del tráfico LAMBS1500, (b) tráfico acumulado y envolvente empírica.

**SOCCKER1500**

Este tráfico es parte del tráfico SOCCER. Se han cogido 1500 tramas siendo un tráfico muy variable (véase la figura A1.7a) con un gran pico sobre la trama 1250. En la figura A1.7b se muestra el tráfico acumulado y la envolvente empírica.

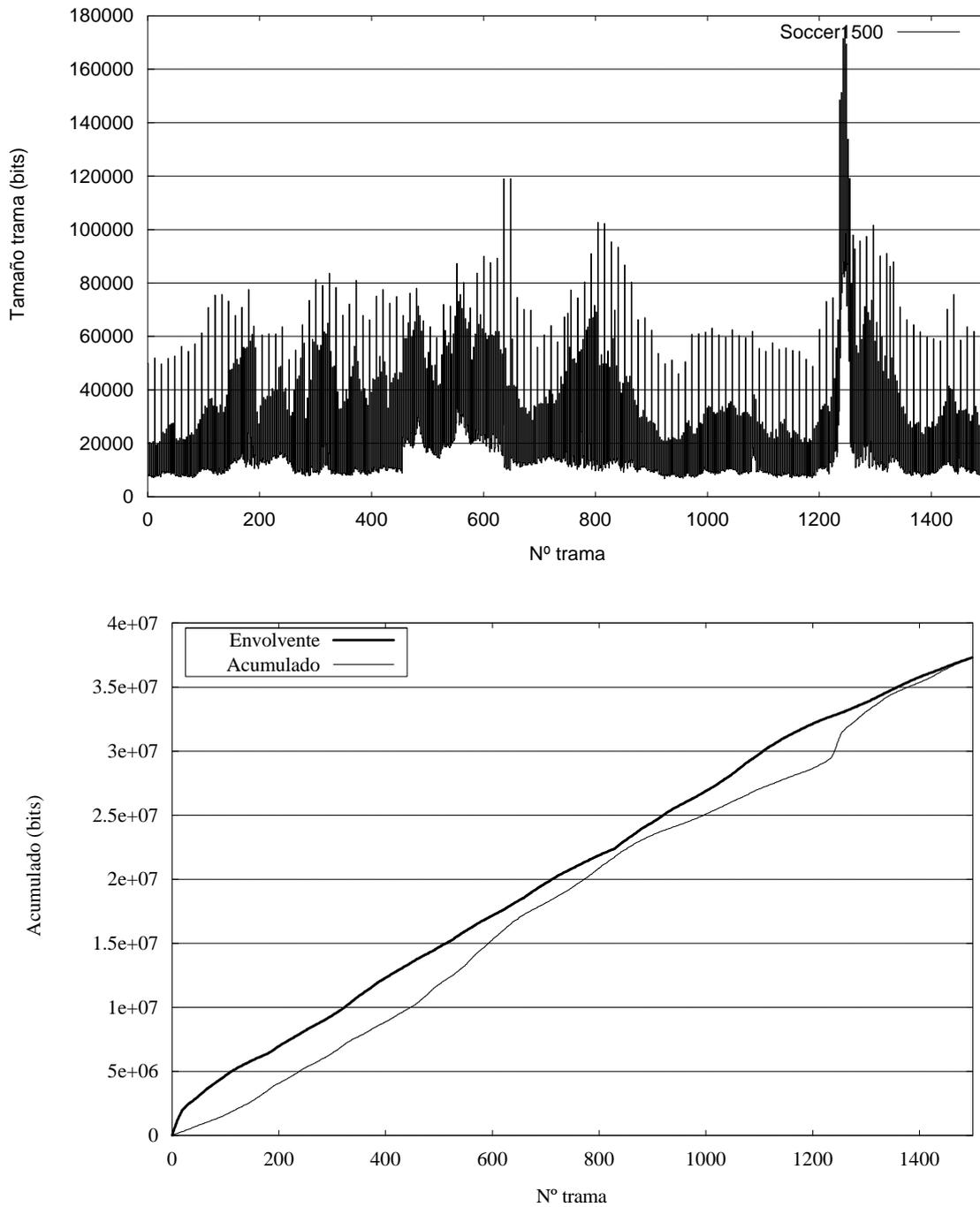


Figura A1.7: (a) Tramas del tráfico SOCCER1500, (b) tráfico acumulado y envolvente empírica.

**NEWS1500**

Este tráfico está compuesto por las 1500 primeras tramas del tráfico **NEWS**. Al igual que el tráfico original tiene un patrón bastante regular, con pocas variaciones. Es el que más se podría acercar a un tráfico CBR.

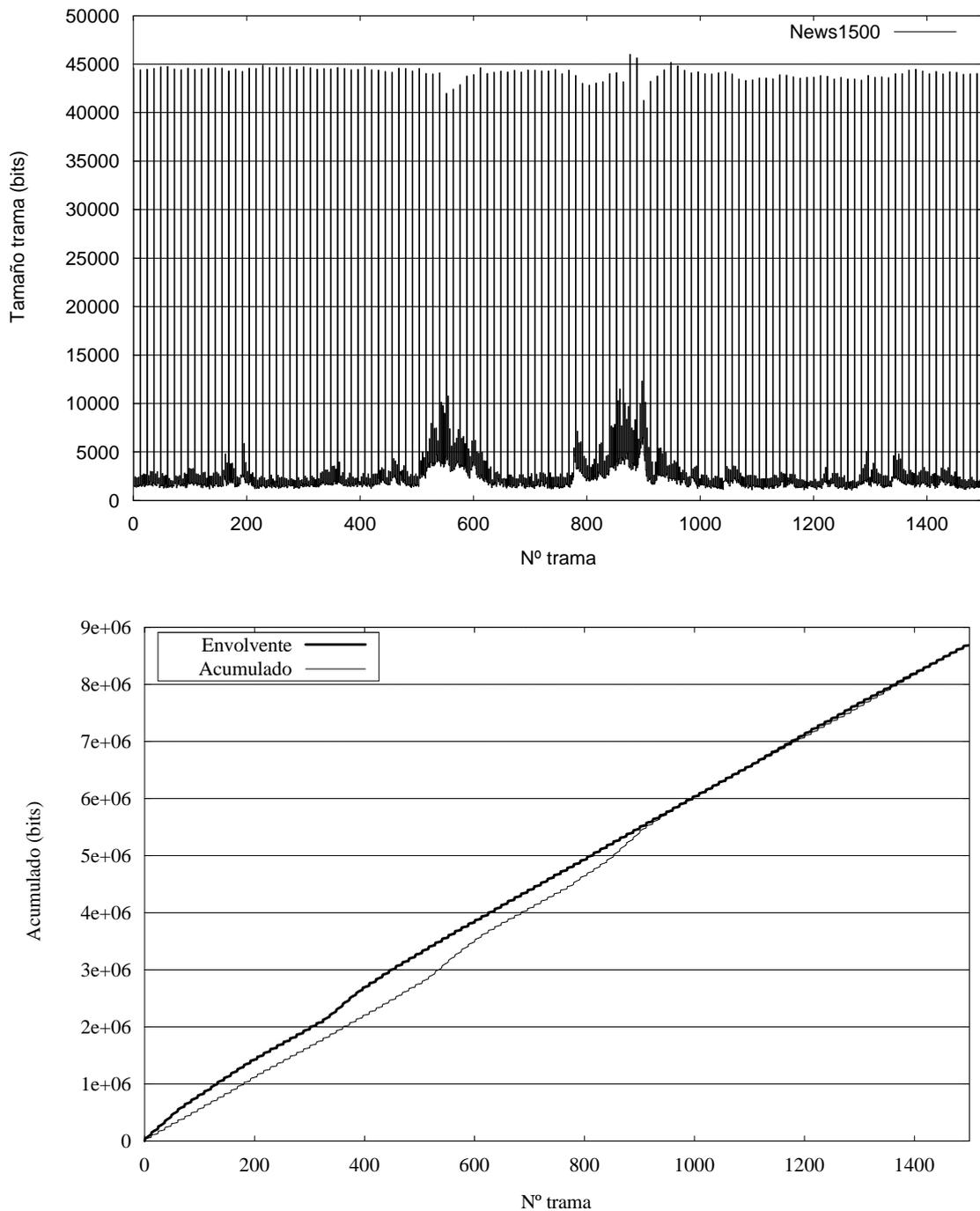


Figura A1.8: (a) Tramas del tráfico NEWS1500, (b) tráfico acumulado y envolvente empírica.

### ***A1.4.- Conclusiones***

En este apéndice se han descrito las trazas MPEG usadas a lo largo de la tesis. Es importante tener una visión del tipo de tráfico que ha sido usado, así como una catalogación del mismo.

También quiero agradecer a O.Rose por poner a nuestra disposición las trazas MPEG utilizadas en su tesis doctoral, lo cual nos ha ahorrado mucho trabajo.

## Apéndice 2: Descripción de RTNetSim

### ***A2.1.- Introducción***

Es necesario poder evaluar y comparar diseños de redes y protocolos antes de su implementación real. Para ello existen técnicas analíticas, simulación y experimentación. El funcionamiento de las redes de comunicaciones puede ser convenientemente descrito por programas de simulación [*Frost94*]. Para ello se requiere:

- modelar las peticiones del usuario de los recursos de la red,
- caracterizar los recursos de la red necesarios para procesar esas peticiones, y
- estimar el rendimiento del sistema basándose en los datos generados por la simulación.

A grandes rasgos, las redes de telecomunicaciones pueden ser vistas como usuarios que generan peticiones a los recursos de la red, y protocolos (algoritmos distribuidos) que controlan la asignación de los recursos para satisfacer esas peticiones. La generación de las peticiones de usuario y su cumplimiento son encapsuladas en eventos de simulación, que son ordenados por el tiempo en que se producen. Este procesamiento basado en eventos conduce a un método conocido como simulación discreta por eventos (DES: *Discrete-Event Simulation*).

En DES, el estado del sistema simulado es almacenado en un conjunto de variables de estado. Las rutinas de evento provocan que las variables de estado se modifiquen. Una lista de eventos se usa para controlar la secuencia de ejecución de estas rutinas de eventos; la lista consiste en eventos ordenados cronológicamente. Las rutinas de evento pueden añadir o eliminar elementos de esta lista de eventos. Simplificando, ejecutar una simulación es, en esencia, la ejecución repetida de un bucle, donde en cada iteración el evento más inminente es ejecutado.

En la simulación de redes de comunicaciones, los elementos con los que se trabaja en las rutinas de evento incluyen llamadas, paquetes y celdas. Cada uno de estos elementos está representado por estructuras de datos u objetos, que aparte de contener la información del sistema a simular, mantienen informaciones estadísticas para la simulación.

Cada sistema DES debe tener un mecanismo de inicialización para establecer el estado inicial del sistema, rutinas de recolección y tratamiento de los datos para obtener medidas, un programa para procesar estas medidas en los parámetros deseados, y un programa que coordine el control de la lista de eventos, el inicio y el fin de la simulación.

Además, el modelado del tráfico es un elemento clave para simular redes de comunicaciones.

## ***A2.2.- Simulador RTNetSim***

**RTNetSim (Real-time Network Simulator)** es un simulador de redes orientado a objetos desarrollado en C++. La simulación se realiza utilizando la librería SMPL adaptada y convertida en una clase C++. El objetivo de este simulador es el de probar el funcionamiento y rendimiento de los protocolos de tiempo real ante determinadas cargas parametrizadas.

**RTNetSim** permite:

- Especificar la red.
- Definir flujos a simular de tipo variable y constantes.
- Implementar y modificar los distintos módulos, como por ejemplo, el planificador.
- Simular.
- Extraer los resultados.

El simulador contiene unos elementos globales genéricos (nodos, terminales, enlaces, etc ) y sobre éstos se han ido implementado los elementos particulares para cada simulación.

Para esta tesis se ha implementado el protocolo WFQ y una parte del esquema de reserva RSVP. Como conformadores de red se han implementado los modelos *leaky bucket*

y *token bucket*. Gracias al diseño orientado a objetos del simulador se podría, por ejemplo, sustituir el protocolo de paquetes por cualquier otro sin que afecte al resto de elementos.

Además, se han implementado en este programa todos los algoritmos desarrollados en esta tesis, como la obtención de los puntos envolventes, optimización de reservas, etc, lo que ha servido para su validación.

El programa está desarrollado en Visual C++ 4.2 y se ejecuta en el entorno Windows, siendo fácilmente adaptable a otros entornos y compiladores. En la figura A2.1 se puede ver un ejemplo de la pantalla del simulador.

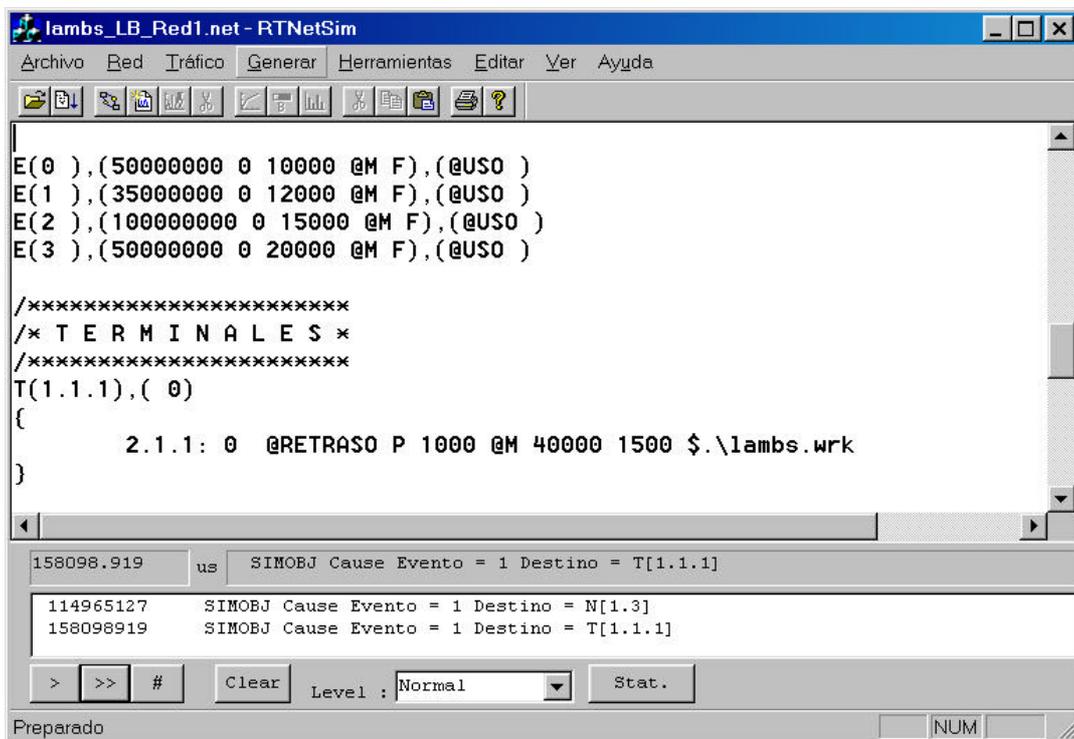


Figura A2.1: Simulador RTNetSim.

### A2.3.- Estructura global de RTNetSim

RTNetSim se compone a grandes rasgos de los siguientes objetos:

- **Terminales:** Son los elementos que envían y reciben los mensajes. Inician la transmisión de un flujo por un canal de acuerdo a la descripción de la carga.
- **Enlaces:** Transmiten los mensajes de un elemento a otro. Permite simular los tiempos de transmisión. Incluye el planificador de paquetes asociado al enlace.

- **Nodos:** Se encargan de recibir un mensaje por el enlace de entrada y transmitirlo al enlace de salida de acuerdo a unas reglas de encaminamiento.
- **Mensajes:** Son la unidad básica de transmisión de los flujos. Están asociados siempre a un canal.
- **Canal:** Define las características de un canal y mantiene la información estadística de la simulación.
- **Planificador:** Planificador de mensajes para decidir que mensaje se envía por el enlace. Se ha implementado el WFQ.

Para el funcionamiento del simulador se tienen además las siguientes clases:

- **CSimObj:** Clase contenedora de todos los objetos y el que realiza la simulación.
- **CGenerador:** Lee el fichero con la configuración de la red, crea todos los objetos y configura la red.
- **Token:** Es la unidad básica de la simulación. Cualquier evento que se produce en el simulador tiene asociado un *token*.
- **Csmpl:** Clase que contiene la librería *simpl*

## A2.4.- Especificación de la red y carga

La red se especifica por medio de un fichero de texto en el que se definen los enlaces, terminales y nodos con el siguiente formato<sup>15</sup>:

```
RED ::= VARIABLE ... ENLACE ... TERMINAL ... NODO
```

Los comentarios comienzan con una / en la primera línea.

### Definición de variables

La sintaxis para definir una variable es:

```
VARIABLE ::= VARIABLE_NORMAL | VARIABLE_ITERATIVA
VARIABLE_NORMAL ::= @Nombre = Valor
VARIABLE_ITERATIVA ::= #Nombre = ( Valor1 Valor2 ... ValorN )
```

---

<sup>15</sup> Se va utilizar un formato de especificación basado en BNF.

donde se describe cada elemento en la tabla A2.1.

Campo	Descripción
Nombre	Nombre de la variable.
Valor	Valor de la variable.
Valor1..n	Valor que toma la variable en la iteración.

Tabla A2.1: Descripción de los campos variables.

Las variables normales definen un valor y se sustituye su valor donde están situadas. Las variables iterativas permiten definir varios valores para una variable y hacer las simulaciones para cada uno de estos valores. Como máximo se pueden definir dos variables iterativas. Esto permite hacer varias simulaciones variando dos parámetros.

Existen unas variables que son usadas para la simulación, que se enumeran en la tabla A2.2.

Variable	Descripción
@DURACION	Duración de la simulación en microsegundos.
@NOMBRE	Nombre de la simulación (a efectos simplemente descriptivos).
@FILEOUT	Fichero donde se guarda el resultado de la simulación en el caso de que haya variables iterativas.
@FILEOUTR	Resultado simulado.
@LEAKYBUCKET	Si es uno indica que el tráfico se introduce en la red con el algoritmo <i>leaky bucket</i> . Si no, es <i>token bucket</i> .

Tabla A2.2: Descripción de las variables predefinidas.

Las variables se usan en la descripción de la red (tanto las normales como las iterativas) como @Nombre.

Ejemplo:

```
@DURACION = 3100000
#USO = ( 0 25 50 75 100 )
#MTU = ( 0 1000 2000 5000 )
```

### Definición de enlaces

La sintaxis para definir un enlace es:

```
ENLACE ::= E(nº),( ab delay mtu M tipo ) [( Utilización) ]
```

donde se describe cada elemento en la tabla A2.3.

Campo	Descripción	Unidad
Nº	Número entero positivo que identifica el enlace.	
AB	Ancho de banda del enlace.	bits/s.
Delay	Retraso fijo del enlace (puede ser cero).	microsegundos
MTU	<i>Minimal Transmission Unit</i> : (Tamaño mínimo del paquete a transmitir).	bits
M	Tamaño máximo del paquete a transmitir.	bits
Tipo	F: <i>Fullduplex</i> H: <i>Halfduplex</i> (No se usa).	
Utilizacion	Indica la utilización del enlace. Este valor se refiere a la utilización del resto del ancho de banda no reservado. Para indicar un uso del 100% de todo el enlace se puede indicar un valor > 100.	%

Tabla A2.3: Descripción de las enlaces.

Se puede definir tantos enlaces como se quiera:

```
E(0 ),(10000000 0 2000 10000 F)
E(1 ),(15000000 50 1000 10000 F)
```

## Definición de terminales

La sintaxis para definir un terminal es:

```
TERMINAL ::= T(dir_ip), (num_enlace)
           [ {
               destino_ip: num_canal dmax DES_CANAL;
               ...
           }
           ]
DES_CANAL ::= DES_CANAL_CTE | DES_CANAL_VBE | DES_CANAL_PAR
DES_CANAL_CTE ::= C R b min_msj max_msj T D
DES_CANAL_VAR ::= V R min_msj max_msj tmin p r ESP_CARGA
DES_CANAL_PAR ::= P min_msj max_msj ESP_CARGA
ESP_CARGA ::= Tcarga N°valores { ( v1, v2,.... vn ) | $FicheroIn }
```

En general, se permite describir un tráfico con flujo constante (*DES\_CANAL\_CTE*), con tasa variable y reserva fija (*DES\_CANAL\_VAR*) y carga variable con reserva calculada en función del retraso (*DES\_CANAL\_PAR*). En la tabla A2.4 se describen los distintos elementos de los terminales. La carga se puede especificar en el propio fichero de definición de la red como una serie de valores o bien se puede indicar que se lea de fichero.

Campo	Descripción	Unidad
dir_ip	Dirección IP del terminal.	En formato b.b.b
num_enlace	Enlace al que está conectado el terminal. Tiene que estar definido en los enlaces.	
destino_ip	Dirección IP del terminal de destino. No chequea si existe.	
num_canal	Identificador del canal. Número positivo único para toda la red.	
R	Ancho de banda a reservar en la red. Parámetro usado por el planificador WFQ.	Bits/s
Dmax	Retraso máximo permitido en el canal.	microsegundos
b	Tamaño del cubo.	Bits
min_msj	Tamaño mínimo del mensaje que se puede transmitir.	Bits
max_msj	Tamaño máximo del mensaje que se puede transmitir.	Bits
T	Periodo de transmisión en flujo constantes.	microsegundos
D	Duración de la transmisión en flujos constantes.	microsegundos
tmin	Periodo mínimo de transmisión en flujo variable.	microsegundos
r	Tasa de transmisión del cubo.	Bits/s
tcarga	Periodo de descripción de la carga variable.	microsegundos
num_valores	Número de valores de la carga.	entero positivo
vi	Tasa de transmisión para el periodo indicado.	bits/s
\$ficheroIn	Fichero donde lee la carga.	

Tabla A2.4: Descripción de los terminales.

Ejemplos :

```
T(1.1.1),( 0)
{
  2.1.1: 0 1000000 C 1650000 500000 750 4500 500 3000000;
  2.1.1: 1 500000 C 5000000 250000 5000 5000 1000 3000000;
  2.1.1: 2 500000 V 1500000 20 10000 1000 1000000 360000
    100000 31
    ( 200000 100000 200000 0 0
      100000 300000 200000 300000 200000
      100000 200000 0 200000 100000
      300000 400000 700000 400000 300000
      500000 600000 500000 600000 800000
      1000000 800000 600000 400000 500000
      200000) ;
  2.1.1: 3 1000000 P 1000 1000 40000 1500 $.\lambdas.wrk
}
```

### Definición de nodos

La sintaxis para definir un terminal es:

```
NODO ::= N(dir_ip), (num_enlace ... ) TABLA_ENRUTAMIENTO
TABLA_ENRUTAMIENTO ::= {
  mascara_ip: num_enlace_salida;
  ...
}
```

Campo	Descripción	Unidad
dir_ip	Dirección IP del nodo.	En formato b.b.b
num_enlace	Enlaces a los que está conectado el terminal. Tiene que estar definido en los enlaces.	
maskara_ip	Mascara IP de enrutamiento. Donde se pone * se selecciona cualquier valor.	
Num_enlace_salida	Enlace por el que saldrá el mensaje si cumple la mascara.	

Tabla A2.5: Descripción de los nodos.

Se pueden definir tantos enlaces como se quiera:

```
N(1.1),(0 1)
{
2.1.* : 1;
1.1.* : 0
}
N(1.2),(1 2)
{
2.1.* : 2;
1.1.* : 1
}
```

A continuación se muestra un ejemplo de la red 1 utilizada en el capítulo 6 con el tráfico LAMBS.

```

/*****
* V A R I A B L E S *
*****/
@NOMBRE = 002VBR100
@DURACION = 4000000
@M = 8192
#RETRASO = (100000 50000 25000 20000 10000)
#USO = ( 0 25 50 75 100)
@FILEOUT = ".\lambs_LB_Red1.RES"
@FILEOUTR = ".\lambs_LB_Red1.RR"
@LEAKYBUCKET = 1

/*****
* E N L A C E S *
*****/
E(0 ),(50000000 0 10000 @M F),(@USO )
E(1 ),(35000000 0 12000 @M F),(@USO )
E(2 ),(100000000 0 15000 @M F),(@USO )
E(3 ),(50000000 0 20000 @M F),(@USO )

/*****
* T E R M I N A L E S *
*****/
T(1.1.1),( 0)
{
2.1.1: 0 @RETRASO P 1000 @M 40000 1500 $.lambs.wrk
}
T(2.1.1),( 3)

/*****
* N O D O S *
*****/
N(1.1),(0 1)
{
2.1.* : 1;
1.1.* : 0
}
N(1.2),(1 2)
{
```

```
2.1.* : 2;  
1.1.* : 1  
}  
N(1.3),(2 3)  
{  
2.1.* : 3;  
1.1.* : 2  
}
```

### ***A2.5.- Estructura de la red***

El simulador ha sido desarrollado para simular el comportamiento de canales en tiempo real en redes configurables. Por esto, se ha diseñado la red basándose en el funcionamiento de RSVP.

Las características de los distintos elementos de la red son los siguientes:

- **Terminal:** El terminal que empieza la conexión se encarga de iniciar el canal y generar el tráfico. En el caso de canales que utilicen RSVP para reservar el canal se usa el protocolo definido: el emisor inicia el canal enviando un mensaje **Path** (como mensaje urgente) para establecer el canal,. cuando lo recibe el terminal de destino comprueba si se puede establecer calculando el valor R y devuelve de vuelta el mensaje **Resv** al origen. Cuando este mensaje llega al emisor se empieza a transmitir la carga.
- **Nodo:** Se encarga de encaminar los paquetes por medio de las tablas de rutas al destino.
- **Enlace:** Hay que aclarar que aquí el concepto de enlace incluye además del enlace entre los nodos, lo que sería su gestión en el nodo (el planificador, etc.). Existen dos tipos de tráfico: de sistema y normal. Existe una única cola para los paquetes de sistema que tienen la máxima prioridad. Si esta cola está vacía el algoritmo de planificación es el WFQ. El enlace se encarga también de gestionar los mensajes **Path** (para actualizar los datos del mensaje con los de enlace) y **Resv** (para reservar el ancho de banda especificado en el planificador WFQ).

### ***A2.6.- Resultados y estadísticas***

El objetivo de toda simulación es obtener unos resultados sobre los cuales extraer conclusiones. En este sentido, las estadísticas que se obtienen están muy orientadas al objetivo de los experimentos necesarios en esta tesis.

En todas las pruebas realizadas, se ha utilizado el tráfico de trazas MPEG de Rose importándolas como un fichero en las simulaciones. A partir de la información generada por las simulaciones se han generado los gráficos y estadísticas insertados en esta tesis usando el programa Gnuplot [Gnuplot98]. Para cada prueba se ha realizado únicamente una simulación al ser determinista. Esto es debido a que se utilizan tráficos fijos y predeterminados con lo que el resultado de las simulaciones son siempre las mismas.

### Información sobre Canales:

Para cada canal se saca la siguiente información:

- Mensajes enviados y recibidos.
- Tiempos medio, máximo y mínimo de transmisión de un mensaje.
- En el caso de conexión RSVP: Tiempo de establecimiento de canal y reserva realizada.
- Para canales con carga variable: Mensajes enviados, recibidos, tiempos medio, máximo y mínimo por intervalo de transmisión.
- Distribución de los mensajes por intervalos de retraso.

Un extracto de la información que se genera aparece a continuación:

```
Canal = 2 Msj Env = 147 Rec = 138 Tmed = 1907.93us Tmax = 7318.87us Tmin = 616.67us
Tiempo establecimiento canal = 1505.00us R = 910620(bits)
Intervalo[ini,fin]      M.Env  Rec      Tmed      Tmax      Tmin
[ 1505, 101505] 101 100 1012.16us 1498.05us 616.67us
[ 101505, 201505] 46 38 4265.19us 7318.87us 1188.17us
Distribución Msj Intervalo[ini,fin] Msj % acumulado
[ 0, 1000] 56 40.58 40.58
[ 1000, 2000] 50 36.23 76.81
[ 2000, 3000] 4 2.90 79.71
[ 3000, 4000] 7 5.07 84.78
[ 4000, 5000] 8 5.80 90.58
[ 5000, 6000] 4 2.90 93.48
[ 6000, 7000] 6 4.35 97.83
```

### Información sobre los Enlaces:

Para cada enlace se saca la siguiente información:

- Mensajes que han pasado por él, bits transmitidos y utilización.
- Para cada canal que pasa por el enlace: Número de mensajes, tiempo medio, máximo y mínimo. Número máximo de mensajes y bits encolados.

Un ejemplo de la información generada es la siguiente:

```
Enlace E2 N° Msj = 9873 N°bits = 40626000 U = 65.49
Canal 3 N° = 900 Tmed = 150.00us Tmax = 150.00us Tmin = 150.00us N°Max = 1 MaxBits = 3000
Canal 2 N° = 2800 Tmed = 150.00us Tmax = 150.00us Tmin = 150.00us N°Max = 1 MaxBits = 1000
Canal 1 N° = 2966 Tmed = 250.07us Tmax = 455.00us Tmin = 250.00us N°Max = 1 MaxBits = 5000
Canal 0 N° = 3207 Tmed = 156.71us Tmax = 238.33us Tmin = 150.00us N°Max = 1 MaxBits = 4498
```

## ***A2.7.- Conclusiones***

En este apéndice se ha descrito brevemente el simulador desarrollado con el objetivo de probar los distintos esquemas y algoritmos desarrollados en esta tesis. El desarrollo del propio simulador ha sido muy útil a la hora de diseñar soluciones a los distintos problemas planteados y la corrección de los algoritmos descritos.

También, como parte de este programa, se han implantado todos los métodos de optimización de reserva de ancho de banda y memoria, permitiendo su comparación con los demás esquemas.

Dado el diseño modular del programa, está abierto a posibles nuevos experimentos, como la prueba de algoritmos de planificación, etc.



## Apéndice 3:

# Modelo de red ORNet

### ***A3.1.- Introducción***

Como compendio de todas las ideas y mejoras introducidas en esta tesis, se plantea el siguiente modelo de red para tiempo real denominado ORNet (*Optimal Reservation Network*). No es el objetivo de esta red su implementación, sino la de servir de base sobre cómo integrar los distintos elementos y esquemas de optimización desarrollados en esta tesis. Las principales características de diseño de esta red son:

- Proporciona retrasos garantizados emisor-receptor.
- Proporciona canales de reserva para obtener una mayor disponibilidad ante fallos de red.
- Optimiza los recursos de red, tanto de ancho de banda como memoria en los nodos.
- Utiliza protocolos y planificadores bien estudiados.
- Permite adaptarse a las variaciones de carga de las subredes.

Para que esta red funcione se necesita un protocolo de reserva de recursos como RSVP y un nivel de red que proporcione transmisión en tiempo real con servicio garantizado. Para implementar este nivel de red se podría utilizar el modelo propuesto por IETF para los servicios integrados que se compone de los siguientes elementos:

- La especificación de tráfico es la de servicio garantizado.
- La disciplina de servicio es WFQ o alguna de sus variantes.
- La conformación de tráfico se realiza en el emisor y es del tipo *token bucket*.

- Esto nos proporciona ecuaciones de retraso y por lo tanto control de admisión.
- El enrutamiento y direccionamiento de paquetes es proporcionado por un proceso enrutador.

El control de admisión y la reserva de recursos la realiza el protocolo RSVP. Se parte por tanto de un esquema como el de la figura 2.12. Por tanto, la arquitectura de la red ORNet se basará en los módulos descritos antes que no necesitan ser modificados ni adaptados, lo cual hace fácil y rápida su implementación.

Como se ha descrito durante la tesis, las distintas optimizaciones se realizan en varios elementos de la red como se detalla en la tabla A3.1.

Componente	Elemento	Descripción
Optimización ancho de banda	Emisor	Optimización del ancho de banda de los enlaces de toda la ruta para obtener un retraso garantizado.
Optimización memoria nodos	Nodos	Optimización de la memoria necesaria en los nodos para no perder ningún paquete agrupando varios canales.
Canales de reserva	Emisor y receptor	Gestión de los canales primarios y reserva. Creación de los canales, optimización de la reserva y activación del canal de reserva.

Tabla A3.1: Optimizaciones en la red.

### ***A3.2.- Arquitectura de la red.***

En este apartado se detalla la arquitectura de la red ORNet. En general, se pueden distinguir tres tipos de nodos en los que se añaden una serie de módulos: emisor, receptor y *router* tal como se detalla en la figura A3.1. Sólo hay que añadir un módulo en el *router* si se requiere optimización de memoria en los nodos o subredes.

El nodo emisor contiene los medios a transmitir y acepta las peticiones de los clientes (receptores). Los elementos que componen el emisor están detallados en la tabla A3.2. El emisor se encarga de aceptar o rechazar las peticiones de los emisores, obtener la reserva óptima a partir del tráfico a transmitir y los parámetros de la red utilizando los puntos envolventes. Si es necesario, gestiona la reserva de recursos para los canales de reserva.

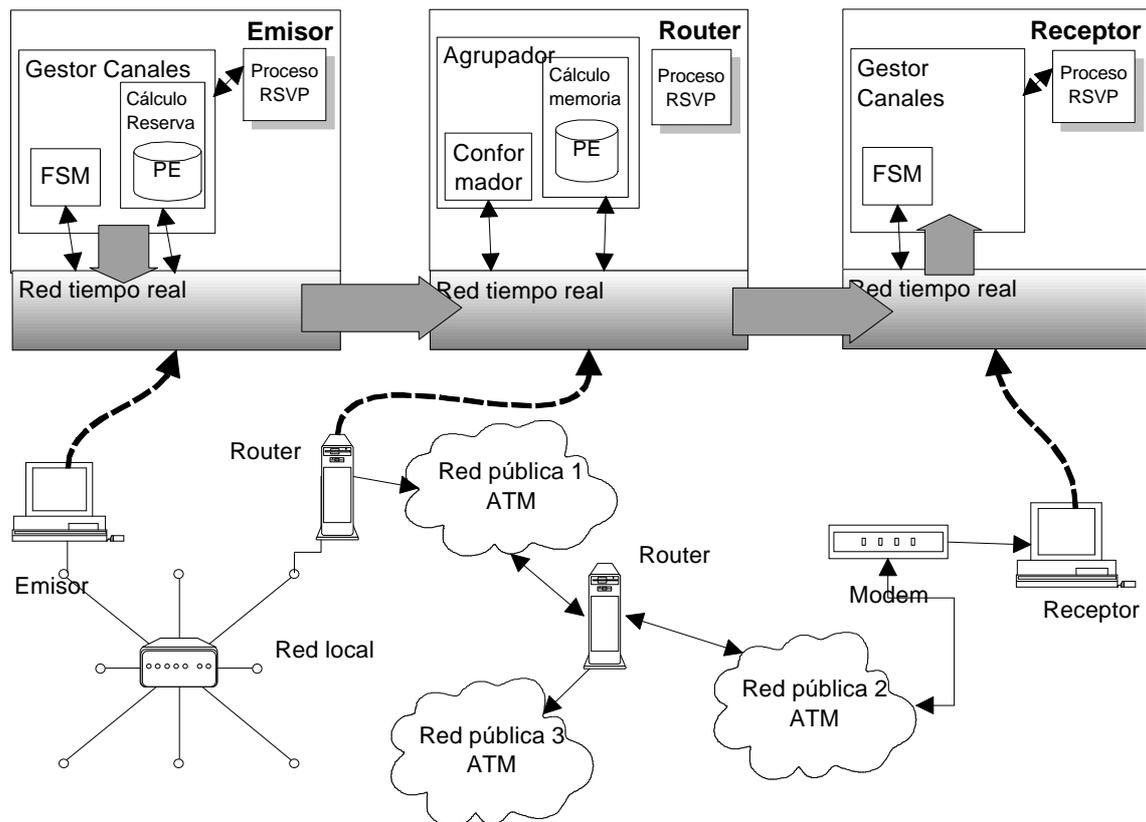


Figura A3.1: Arquitectura de la red ORNet.

En el caso de que se quiera optimizar la memoria en los nodos, el emisor tiene que enviar antes del comienzo de la transmisión un paquete especial conteniendo los puntos envolventes. Como se ha visto en la tesis, los puntos a enviar son pocos por lo que este paquete no sería demasiado grande. Este paquete sería capturado por los *routers* que lo necesitan y estaría asociado al canal a transmitir.

Componente	Descripción
Gestor Canales	<i>Middleware</i> que proporciona un API a la aplicación y gestiona los distintos módulos en función del servicio requerido.
Cálculo reserva	A partir de los Puntos envolventes y los parámetros de red obtiene la reserva óptima aplicando el algoritmo desarrollado en el capítulo 4.
FSM	Módulo que proporciona el soporte a canales de reserva.

Tabla A3.2: Componentes del emisor.

El receptor es más simple ya que no contiene el módulo de reserva óptima. Los elementos que lo componen están en la tabla A3.3. El receptor iniciará la conexión con el emisor pidiendo una determinada calidad de servicio. Esto provocará el establecimiento de un canal entre emisor y receptor, y en el caso de que el canal sea admitido la reserva de los

recursos en la red y el comienzo de la transmisión. El receptor es también responsable, en el caso de que la aplicación lo requiera, de la gestión de los canales de reserva. El módulo FSM pedirá a la red la creación de otro canal entre emisor y receptor con el retraso necesario y controlará la recepción de mensajes para comprobar que llegan antes del tiempo de fallo  $d_f$ .

Componente	Descripción
Gestor Canales	<i>Middleware</i> que proporciona un API a la aplicación y gestiona los distintos módulos en función del servicio requerido.
FSM	Módulo que proporciona el soporte a canales de reserva.

Tabla A3.3: Componentes del receptor.

La implementación del *router* dependerá de las características necesarias. Si no se optimiza la memoria en los nodos o subredes no será necesario añadir ningún módulo en los *routers*. En el caso de que queramos optimizar la memoria al entrar en una subred agrupando los canales en un macrocanal, se puede añadir el módulo agrupador.

Este módulo agrupador estaría antes del planificador del nodo y comprobaría el estado de los paquetes. Para ello, obtendría del primer paquete especial de la transmisión los puntos envolventes y los guardaría para el cálculo de la memoria. La subred tendría que informar del ancho de banda disponible para el macrocanal con el objeto de calcular las nuevas reservas individuales de ancho de banda para cada canal.

Este módulo además, interceptará los mensajes asociados a un macrocanal y los pasará por los conformadores de tráfico *leaky bucket multinivel* antes de enviarlos al planificador.

Componente	Descripción
Agrupador	Módulo que agrupa los canales en función de las direcciones destino a la entrada de una subred.
Cálculo de memoria	Calcula la memoria óptima necesaria a partir de los puntos envolventes y el ancho de banda disponible para el macrocanal.
Conformador	Conforma el tráfico que está agrupado en un macrocanal, siguiendo el modelo <i>leaky bucket multinivel</i> .

Tabla A3.4: Componentes de un router con optimización de memoria.

### A3.3.- Interfaz de programación

En este punto se describe un ejemplo de interfaz de programación o API que podría proporcionar la red ORNet a las aplicaciones. Este API sería de muy alto nivel y encapsularía otros protocolos de bajo nivel. Con pocas funciones se dispondría de un servicio de transmisión de ficheros en tiempo real, lo cual simplificaría la implementación

de vídeo bajo demanda. Sin entrar en detalle sobre los parámetros, las funciones necesarias están descritas en la tabla A3.5.

Aparte de la simplificación de uso, otra ventaja añadida de este API es que no hay que especificar el flujo. Las características del flujo las calculará directamente el gestor de canales a partir de los puntos envolventes que se han generado a partir del tráfico.

Función	Descripción
IdCanal OR_PideFichero(Origen, Fichero, QoS, Callback)	El cliente pide un fichero a un servidor determinado por la dirección Origen con una determinada calidad de servicio. Además se añade una función <i>callback</i> para que cuando empiece la transmisión reciba los paquetes de la transmisión.
IdCanal OR_EsperaPetición(Origen, Fichero, QoS)	El servidor llama a esta función que espera peticiones por parte de los clientes. De esta forma el servidor puede aceptar o rechazar una determinada petición en función del origen o el tipo de fichero pedido. Esta función sólo devuelve el control a la aplicación llamante en el caso de que el canal pueda ser establecido. Es decir, cuando llega el mensaje Path de RSVP.
BOOL OR_CalculaReserva(IdCanal, Puntos, QoS)	Esta función calcula primero si se puede admitir el nuevo canal utilizando el control de admisión. Después calcula con los puntos envolventes y los parámetros de red la reserva óptima a partir del retraso exigido.
OR_PeticiónDenegada(IdCanal, Causa)	Indica al cliente que la petición ha sido denegada, ya sea por cuestiones de aplicación (permisos, saldo insuficiente, etc) o bien por fallar el control de admisión. Esto generará un mensaje RSVP ResvErr.
OR_TransmiteFichero(IdCanal, Fichero, QoS)	Si el servidor acepta la petición utilizará esta función para transmitir el fichero. Lo único que se indica es el destino y el fichero. El gestor ya se encargará de ir transmitiendo el fichero. Esto provoca la generación del mensaje RSVP Resv de vuelta al cliente. Además se envía un primer paquete especial con los puntos envolventes.
OR_CancelaTransmisión(IDCanal)	Cancela la transmisión. Envía un mensaje RSVP PathTear al emisor.
OR_CalculaPE(Fichero, Puntos)	Función auxiliar que calcula los puntos envolventes de un tráfico. Se utilizará <i>off-line</i> para obtener los puntos envolventes cuando se introduce un nuevo vídeo en el servidor.

Tabla A3.5: API de ORNet.

### A3.4.- Comportamiento dinámico

En este punto se muestra un ejemplo del comportamiento dinámico de esta red. La secuencia de llamadas de un cliente que se conecta a un servidor de vídeos está detallado en la figura A3.2:

Receptor (cliente)	Emisor (servidor VoD)
Id = OR_PideFichero(...,FnCB)	
.	
.	
.	Id = OR_EsperaPetición(...)
.	CreaHilo(Trata_petición)
.	...
.	Trata_petición(Id) {
.	<b>if</b> (CompruebaPetición(...) <b>and</b>
.	OR_CalculaReserva(...))
.	{
.	OR_TransmiteFichero(...)
.	}
.	<b>else</b>
.	{
.	OR_DeniegaPetición(...)
.	}
.	}
<b>if</b> (Id != 0)	
{	
// Canal establecido	
// Recibe paquetes en	
// función FnCB	
}	
FnCB(...)	
{	
// Reproducir el vídeo.	
}	

Figura A3.2: Secuencia de llamadas.

Con más detalle el funcionamiento es como sigue:

1. El cliente, que es el receptor, se conectaría primero a la página web del servidor para elegir que vídeo quiere ver.
2. Decidido el vídeo, el cliente empieza el establecimiento del canal por medio de la llamada de `OR_PideFichero` indicando los parámetros de calidad requeridos.
3. El servidor tiene un hilo (*thread*) esperando peticiones, esto es, bloqueado en la función `OR_EsperaPetición`.
4. Cuando el servidor recibe la petición del cliente por medio del mensaje Path de RSVP, este hilo se desbloquea y crea otro hilo para la gestión de la nueva petición.

5. Este nuevo hilo, primero comprueba que se puede aceptar la petición realizando una serie de comprobaciones (identificación de cliente, seguridad, saldo disponible, etc.).
6. Si se admite la petición, se buscarán los puntos envolventes asociados al vídeo pedido y se llamará a la función `OR_CalculaReserva` para comprobar si la red puede admitir el canal y en caso positivo optimizar la reserva.
7. En caso negativo, se llamará `OR_PeticionDenegada` indicando la causa del rechazo de la petición (para toda esta gestión se utilizarán los mensajes RSVP apropiados).
8. En caso positivo, se empieza la transmisión del vídeo por medio de la función `OR_TransmiteFichero`.
9. Esto provoca, el envío del mensaje de RSVP `Resv` de vuelta al origen para confirmar la reserva de recursos en la red.
10. Cuando el cliente recibe este mensaje la función `OR_PeticionDenegada` devuelve el control a la aplicación con lo que el canal está establecido y la transmisión comienza. Cada vez que el cliente reciba un paquete se invocará a la *callback* indicada en la función.

El establecimiento de los canales de reserva es totalmente transparente a las aplicaciones cliente y servidor. En el caso de que el cliente pida este servicio entrarían en juego los módulos FSM del emisor y receptor. El funcionamiento de este módulo está descrito en el punto 6.6.

La optimización de la memoria en la subred se realiza en el *router*. Como se ha comentado, en el comienzo de la transmisión de un canal el primer paquete que se envía contiene los puntos envolventes del tráfico que se va a transmitir. La información de este paquete será recogida por los *routers* con un módulo agrupador. Cuando este paquete llegue al receptor será ignorado.

Tal como se comentó en el punto 5.7.1 estos *routers* agruparán el tráfico con un mismo destino dentro de la subred y negociará un ancho de banda para este macrocanal. Además, este mecanismo es dinámico en ambos sentidos, es decir, se pueden añadir o eliminar canales a un macrocanal y se puede variar el ancho de banda disponible para el canal.

Estas variaciones implicarán un recálculo de la reserva de ancho de banda individual de cada canal, lo cual se notificará a la subred.

### ***A3.5.- Conclusiones***

En este apéndice se ha descrito un modelo de red ORNet que detalla como implementar las optimizaciones desarrolladas en esta tesis. Este modelo es fácil de implementar debido a que parte de la base de una red de transmisión en tiempo real y se limita a añadir una capa intermedia o *middleware* que gestiona el nivel de red y el protocolo RSVP.

También se ha descrito un prototipo de API que proporcionaría este *middleware* el cual simplificaría enormemente la implementación de servidores de vídeo bajo demanda.

## Apéndice 4: Acrónimos

AAL: *ATM Adaptation Layer*. (ATM)  
AB: Ancho de banda.  
ABR: *available bit-rate traffic*. (ATM)  
ADPCM: *Adaptative Differential Pulse Code Modulation*.  
API: *Application Programming Interface*.  
ATM: *Asynchronous Transfer Mode*.  
CAC: *connection admission control*). (ATM)  
CBO: *continuous bit-stream oriented traffic*.  
CBR: *Constant Bit Rate*.  
CCITT: Comité Consultatif International de Télégraphique et Téléphonique.  
CELP: *Code excited linear predictor*.  
CER: *cell error ratio*.  
CLR: *cell loss ratio*. (ATM)  
CMR: *cell misinsertion rate*. (ATM)  
CTD: *cell transfer delay*. (ATM)  
DCT: *Discrete Cosine Transform*.  
DES: *Discrete Event Simulation*.  
DVI: *Digital Video Interactive*.  
EDD: *Earliest Due Date*. (Disciplina de servicio).  
FDDI: *Fiber Distributed Data Interface*.  
FFQ: *Fluid Fair Queuing*. (Disciplina de servicio).  
FGN: *Fractional Gaussian Noise*.  
FTP: *File Transfer Protocol*.  
GOP: *Group of pictures*.(MPEG).  
GPS: *Generalized Processor Sharing*. (Disciplina de servicio).  
IETF: *Internet Engineering Task Force*.  
IPX: *Internetwork Packet Exchange*.  
ISO: *International Standard Organization*.

ITU : *International Telecommunications Union.*  
JPEG: *Joint Photographic Expert Group..*  
LAN: *Local Area Network*  
LBAP: *Linear Bounded Arrival Process*  
LLC: *Logical Link Control.*  
MAC: *Medium Access Control.*  
Max-CTD: *maximum cell transfer delay. (ATM)*  
MBS: *maximum burst size. (ATM)*  
MCR: *minimum cell rate. (ATM)*  
MFS: *maximum frame size. (ATM)*  
MPEG: *Moving Pictures Expert Group.*  
MTU: *Maximum Transmission Unit.*  
NFS: *Network File System.*  
OSI: *Open Systems Interconnection.*  
P2P-CDV: *peak-to-peak cell delay variation . (ATM)*  
PCR: *peak cell rate. (ATM)*  
PGPS: *Packet General Processor Sharing.*  
QoS: *Quality of Service.*  
RCSP: *Rate Controlled Static Priority. (Disciplina de servicio).*  
RDSI: (ISDN) *Red Digital de Servicios Integrados.*  
RFC: *Request for comments.*  
RSVP: *ReSerVation Protocol.*  
RTP: *Real-Time Protocol.*  
RTSP: *Real-Time Streaming Protocol.*  
SCFQ: *Self clocked fair Queueing. (Disciplina de servicio).*  
SCR: *sustainable cell rate. (ATM)*  
SECBR: *severely errored cell block ratio. (ATM)*  
SNA: *Systems Network Architecture.*  
STB: *Set Top Box.*  
TCP/IP: *Transmission Control Protocol / Internet Protocol.*  
TFF: *Tasa de falsos fallos.*  
UBR: *Unspecified Bit Rate. (ATM)*  
UMTS: *Universal Mobile Telecommunication System.*  
VBR: *variable bit-rate traffic.*  
VCI: *virtual circuit identifiers. (ATM)*  
VoD: *Video on Demand.*  
VPI: *virtual path identifiers. (ATM)*  
WAN: *Wide area network.*  
WFQ: *Weighted Fair Queuing. (Disciplina de servicio).*

## Bibliografía

- [Anderson92] P. Anderson, Y. Osawa and Ramesh Govindan, "A file system for continuous Media", *ACM Transactions on Computer Systems*, 10(4):311-337, Nov. 1992.
- [Baker91] J. Baker, "A distributed link restoration algorithm with robust replanning", *In Proc. IEEE GLOBECOMM*, pp. 306-311, 1991.
- [Banerjea93] A. Banerjea, C. Parris and D. Ferrari, "Recovering guaranteed performance service connections from single and multiple faults", *Technical Report TR-93-66*, Int. Computer Science Institute, Berkeley, CA.
- [Banerjea96] A. Banerjea, D. Ferrari, B. Mah, M. Moran, D. Verma and H. Zhang, "The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences", *IEEE/ACM Transactions on Networking*, vol. 4, n. 1, pp. 1-10, February 1996.
- [Beran95] J. Beran, R. Sherman, M.S. Taqqu, and W. Willinger, "Long-Range Dependence in Variable-Bit-Rate Video Traffic", *IEEE Transactions on Communications*, 43(2/3/4):1566-1579, 1995.
- [Bernet00] Bernet et al., "A Framework for Integrated Services Operation over DiffServ Networks", IETF draft, May. 2000. En: <http://www.ietf.org/internet-drafts/draft-ietf-issll-diffserv-rsvp-05.txt>.
- [Blake98] S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang and W. Weiss. "An architecture for Differentiated Services", IETF RFC2475, Dec. 1998.
- [Braden94] R. Braden, D. Clark, S. Shenker, "Integrated services in the Internet Architecture : An Overview". *Internet RFC 1633*. July 1994.
- [Braden97] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin : "Resource Reservation Protocol (RSVP). Versión 1 Functional Specification. . *Internet RFC 2205*, Sep. 1997.
- [Braun00] T. Braun, H.J. Einsiedler, M. Schieidegger and G. Stattenberger. "A Linux Implementation of a Differentiated Services Router". *5th IFIP TC6 International Symposium : Interworking'2000 : Network and services for The Information Society. Bergen, Norway, Oct. 2000. In Lecture Notes in Computer Science n° 1938*. Springer-Verlag 2000.

- [Casilari99] E. Casilari, A.Reyes, A. Díaz Estrella y F. Sandoval, “Modelado Wavelet de tráfico de vídeo VBR”. *JITEL'99* 200-206, Sep-1999
- [Chang94] E. Chang and A. Zakhor, “Admissions Control and Data Placement for VBR Video Servers”, In *1<sup>st</sup> IEEE International Conference on Image Processing*, pp. 278-288, Austin, Nov. 1994.
- [Chen97] J.J. Chen and D. W. Lin, “Optimal bit allocation for coding video signal over ATM Networks”, *IEEE Journal. Selected Areas Communication.*, vol. 15, pp. 1002-1015, Aug. 1997.
- [Chiariglione98] L. Chiariglione, “Impact of MPEG Standars on Multimedia Industry”, *Proceedings of the IEEE*, Vol. 86, N° 6, June 1998.
- [Clark92] D. Clark, S. Shenker, and L. Zhang, “Supporting Real-Time Applications in an Integrated Services Packet Network : Architecture and Mechanisms”, In *Proceedings of ACM SIGCOMM'92*, pag. 14-26, Baltimore, Maryland, August 1992.
- [Cobb97] J.A. Cobb, “Flow Theory”, *IEEE/ACM Transactions on Networking*, Vol. 5, N° 5, Oct. 1997, pp. 661-674
- [Crowcroft95] J. Crowcroft, Z. Wang, A. Smith, and J. Adams. A Rough Comparison of the IETF and ATM Service Models. *IEEE Network*, Dec. 95
- [Crowcroft99] J. Crowcroft, M. Handley and Ian Wakeman, *Internetworking Multimedia*. Morgan Kaufmann Publishers, 1999.
- [Cruz91a] R. Cruz, “A calculus for network delay, part I : Network elements in isolation”. *IEEE Transactions on Information Theory* , 37(1):114-131, January 1991.
- [Cruz91b] R. Cruz, “A calculus for network delay, part II : Network analysis”. *IEEE Transactions on Information Theory* , 37(1):132-141, January 1991.
- [Cruz95] R. Cruz, “Quality of Service Guarantees in Virtual Circuit Switched Networks”. *IEEE Journal of Selected Areas in Communications*, Vol. 13, N° 6, pp. 1048-1058, Aug 1995.
- [Deloddere94] D. Deloddere, W. Verbiest and H. Verhille, “Interactive Video On Demand”. *IEEE Communications Magazine*, May 1994 pp. 82-88
- [Demers89] A. Demers, S. Keshav and S. Shenker, “Analysis and Simulation of a Fair Queueing Algorithm”, in *Internetworking : Research and Experience*, Vol 1, No. 1., pp. 3-26. October 1990. También en *Proceedings of ACM SIGCOMM'89*, pp 3-12.
- [Eco97] Umberto Eco, *Como se fu una tesi di laurea*. Tascabili Bompiani, 1977. (trad. española, *Como se hace una tesis*. Ed. Gedisa, 1999).
- [Elwalid95] A. Elwalid, D. Mitra and R. Wentworth, “A new approach for Allocating Buffer and Bandwidth to Heterogeneous, Regulated Traffic in an ATM node”, *IEEE Journal on Selected Area In Communications*, Vol. 13, N°6, pp. 1115-1127, August 1995.

- [Ferrari89] D. Ferrari, "Real-Time Communication in Packet-Switching Wide-Area Networks". *Technical Report TR-89-022. International Computer Science Institute*, Berkeley, May 1989.
- [Ferrari90a] D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide-area networks", *IEEE Journal on Selected areas in Communications*, 8(3):368-379, April 1990.
- [Ferrari90b] D. Ferrari, "Client requirements for Real-time communication services". *RFC 1193*, November 1990.
- [Ferrari91] D. Ferrari, "Distributed Delay Jitter Control in Packet-Switching Internetworks", *Journal of Internetworking: Research and Experience*, vol. 4, n. 1, pp. 1-20, 1993.
- [Ferrari92a] D. Ferrari, "Real-Time Communication in an Internetwork", Technical Report TR-92-001, *Journal of High Speed Networks*, vol. 1, n. 1, 1992, pp. 79-103, 1992.
- [Ferrari92b] D. Ferrari, "Design and Applications of a Delay Jitter Control Scheme for Packet-Switching Internetworks", *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video*, Springer-Verlag, Heidelberg, Germany, pp. 72-83, November 1991; also *Computer Communications*, vol. 15, n. 6, pp. 367-373, July-August 1992.
- [Floyd95] S. Floyd and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", *IEEE/ACM Transactions on Networking*, Vol. 3 no. 4, pp. 365-386, August 1995.
- [ForumATM96] The ATM forum technical Committee, "User to Network Interface (UNI) Signalling Specificatio, version 4.0, af-sig-0061.00 1996.
- [Frost94] V.S. Frost and B. Melamed, "Traffic Modeleling For Telecommunications Networks", *IEEE Communications Magazine*, March 1994, pp. 70-81.
- [Gall91] D.L.Gall, "MPEG: A video compression standard for multimedia applications", *Communications of the ACM*, 34(4):46-58, Apr. 1991.
- [Giroux99] N.Giroux, S. Ganti, *Quality of Service in ATM networks : state of the art traffic management*. Prentice Hall, 1999.
- [Gnuplot98] D. Crawford et al. *Gnuplot : An Interactive Plotting Program*. Version 3.7. User manual, 3 december 1998. En <http://www.gnuplot.org>.
- [Goderis00] D. Goderis et al., "TEQUILA: Functional Architecture Definition and Top Level Design ", Report. Sep.2000. En <http://www.ist-tequila.org/>
- [Golestani90] S.J.Golestani, "A stop-and-go queueing framework for congestion management". In *Proceedings of ACM SIGCOMM'90* pág. 8-18. Philadelphia Pennsylvania, Sept. 1990.
- [Grover87] W. Grover, "The selfhealing network: A fast distributed restoration technique for networks using digital crossconnect machines". In *Proceedings IEEE GLOBECOM* .pág. 1090-1095. 1987.

- [Han97] S.Han and K.G.Shin, "Eficcient Spare-Resource Allocation for Fast Restoration of Real-Time Channels from Network Component Failures", *IEEE* 1997,99-108.
- [Han99] S. Han and K.G. Shin, "Experimental Evaluation of Behavior based Failure-Detection Schemes in Real-Time Communication Networks", To appear in *IEEE Transactions In Parallel and Distributed Computing*.
- [Hehmann90] D.Hehmann, M.Salmony, H.J. Stüttgen, "Transport Services for Multimedia Applications on Broadband Networks", *Computer Comm. Rev.*, Vol 13, N° 4, May 1990, pp. 197-203.
- [Heinanen99] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski. "Assured forwarding PHB group". IETF RFC2597, Jun. 1999.
- [Hernandez00a] E. Hernández, J.Vila, "A fast method to optimise network resources for Video-on-demand Transmission". *IEEE Proceedings of Euromicro 2000* , pp. 440-447., Maastricht, The Netherlands. Sep. 2000
- [Hernandez00b] E. Hernández, J.Vila, "Evaluation of a new resource reservation scheme for MPEG transmission in highly available real-time channels". *5<sup>th</sup> IFIP TC6 International Symposium : Interworking'2000*. Bergen, Norway, Oct. 2000. In Springer Verlag Lecture Notes in Computer Science n° 1938. pp.122-136.
- [Hernandez98a] E. Hernández, "Transmisión de datos en tiempo real : Síntesis de protocolos y redes para transmisión en tiempo real". Memoria trabajo 2 créditos doctorado. Jul. 1998
- [Hernandez98b] E. Hernández, "Control de la calidad del servicio en redes en tiempo real".Memoria trabajo 6 créditos doctorado. Oct. 1998
- [Hernandez99a] E. Hernández, J.Vila, "Reserva eficiente de recursos para canales de reserva en redes en tiempo real", *II Jornadas de Ingeniería Telemática JITEL'99*, Madrid, Sep. 99. Libro de ponencias pp. 11-18.
- [Hernandez99b] E. Hernández, J.Vila, "An efficient Resource Reservation Scheme for Highly available real-time Channels", *Proceedings of PDCS'99, Nov. 1999*, M.I.T., Cambridge, USA. pp. 385-391.
- [Heyman96] D. P. Heyman and T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic", *IEEE/ACM Transactions on Networking*, Vol. 4, N° 1, Feb 1996.
- [Hsu97] C.Y. Hsu, A.Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints" *IEEE Journal Selected Areas Communication.*, vol. 15, pp. 1016-1028, Aug. 1997.
- [Huidobro99] J.M. Huidobro, *Redes y Servicios de telecomunicaciones*. Ed. Paraninfo, 1999.
- [Hurst51] H.E. Hurst, "Long-Term Storage of Reservoirs", *Trans. Amer. Soc. Civil Engineers*, vol 116, 1951, pp. 770-799.

- [Jacobson99] V. Jacobson, K. Nichols and K. Poduri. "An expedited Forwarding PHB". IETF RFC2598, Jun. 1999.
- [Kalmanek90] C.R.Kalmanek, H. Kanakia and S. Keshav. "Rate controlled servers for very high-speed networks". In *IEEE Global Telecommunications Conference*, pág 300.3.1 – 300.3.9, San Diego, California, Dic. 1990.
- [Kao94] B. Kao, H. García-Molina, and D. Barbara. "Aggressive transmissions of short messages over redundant paths". In *IEEE Transactions Parallel and Distributed Systems*, vol. 5, nº 1, pp. 102-109, Jan. 1994.
- [Kelly91] Kelly, F.P, "Effective Bandwidths at Multi-Class Queues", *Queueing Systems*, vol 9 (1991),5-16.
- [Knightly94] E.W.Knightly, R.F.Mines, H. Zhang, "Deterministic Characterization and Network Utilizations for Several Distributed Real-time Applications", In *Proceedings of IEEE WORDS'94*
- [Knightly95] E.W.Knightly, H. Zhang, "Traffic Characterization and switch Utilization using a Deterministic Bounding Interval Dependent Traffic Model", In *Proceedings of IEEE INFOCOM'95*
- [Koch00] B. F. Koch et al. "Aquila : Adaptative Resource Control for QoS Using an IP-based Layered Architecture : Project presentation" Report, Abril 2000. En <http://www-st.inf.tu-dresden.de/aquila/>.
- [Kuznetsov99]A. Kuznetsov, sitio FTP para los fuentes RSVP y del paquete iproute2+tc, URL <ftp://ftp.inr.ac.ru/ip-routing>.
- [Kvols92] K. Vols and S. Blaabjerg, "Bounds and approximations for the periodic on/off queue with applications to ATM traffic control", In *Proceedings of IEEE INFOCOM'92*, pp. 487-494.
- [Kweon98] S. Kweon and K.G.Shin, "Providing Deterministic Delay Guarantees in ATM Networks", In *IEEE/ACM Transactions on Networking*, Vol.6 Nº 6, Dec. 1998, pp. 838-850.
- [Kweon99] S. Kweon and K.G.Shin, "Transport of MPEG Video with Statistical Loss and Delay Guarantees in ATM Networks Using a Histogram-Based source model ", *To be published in IEEE*, 1999.
- [Lakshman98]T.V. Lakshman, A. Ortega and A.R. Reibman, "VBR Video: Tradeoff and Potentials". *Proceedings of the IEEE*, Vol. 86, Nº 5, May 1998.
- [Leland94] W.E.Leland, M.S.Taqqu, W.Willinger, D.V.Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", *IEEE/ACM Transactions on Networking*, Vol, 2, Nº 1, Feb. 1994. pp. 1-15.
- [Lucantoni94] D.M. Lucantoni, M. F. Neuts and A.R. Reibman, "Methods for Performance Evaluation of VBR Video Traffic Models". *IEEE/ACM Transactions on Networking*, Vol. 2, Nº 2, April 1994
- [Luo97] W. Luo and M.E. Zarki, "Quality Control for VBR Video over ATM Networks", *IEEE J. Select. Areas Communication.*, vol. 15, nº 6, pp. 1029-1039, Aug. 1997.

- [Makaroff97] D. Makaroff, G. Neufeld and N. Hutchinson, "An evaluation of VBR Disk Admission Algorithms for Continuous Media File Serves", *Fifth ACM International Multimedia Conference*. 9-13 Nov. 1997.
- [Manzoni99] P. Manzoni, P. Cremonesi and G. Zerazzi, "Workload Models of VBR Video Traffic and Their Use in Resource Allocation Policies", *IEEE/ACM Transactions on Networking*, Vol. 7, N° 3, June 1999.
- [McDysan98] D. McDysan and D. Spohn, "ATM: Theory and Applications", *McGraw Hill Series on Computer Communications*. 1998.
- [McManus96] J. M. McManus and K.W. Ross, "Video-on-Demand Over ATM: Constant-Rate Transmission and Transport". *IEEE Journal of Selected Areas in Communications*, Vol. 14, N° 6, pp. 1087-1098, Aug 1996.
- [Microsoft99] "An overview of QoS". White Paper. Windows 2000 Server Technical Notes. *Microsoft TechNet* – December 1999.
- [Nichols98] Nichols, Blake, Baker and Black "Definition of the Differentiated Services Field (DS Field) in the IPv4 and the IPv6 Headers", *IETF RFC 2474*, Dec.1998
- [Nichols99] K. Nichols, V. Jacobson and L. Zhang. "A Two-bit differentiated Services Architecture for the Internet". *IETF RFC2638*, Jul. 1999.
- [Pancha94] P. Pancha and M.E. Zarki, "MPEG Coding for variable bit rate video Transmission", *IEEE Communications Magazine*., pp. 54-66, May 1994.
- [Parekh92] A.K.Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case", *IEEE/ACM Transaction on Networking*, vol. 1, n° 3, pp. 344-357, June 1993
- [Parekh93] A.K.Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case", *IEEE/ACM Transactions on Networking*, vol. 2, n° 2, pp. 137-150, April 1994
- [Partridge92] C. Partridge, "A Proposed Flow Specification", *RFC 1363*, September 1992.
- [Peña00] O. Peña, *Multimedia: Edición 2000*. Guías prácticas Anaya. 2000.
- [Presti99] F.L. Presti, Z.L. Zhang, J. Kurose, and D. Towsley "Source Time Scale and Optimal Buffer/Bandwidth Trade-off for Heterogeneous Regulated Traffic in a Network Node" *IEEE Transactions on Networking*, Vol 7, N°4 , August 1999
- [Ramanathan92] P.Ramanathan and K.G.Shin, Delivery of Time-Critical Messages Using a Multiple Copy Approach. *ACM Transaction on Computer Systems*, Vol. 10, No. 2, May 1992, 144-166.
- [Ranga91] P. V. Rangan and H.M.Vin, "Designing File Systems for Digital Video and Audio". In *Proceedings 13<sup>th</sup> Symposium on Operating Systems Principles (SOSP'91)*, Operating Systems Review, volume 25, pages 81-93, October 1991.

- [Rexford99] J. Rexford and D. Towsley, "Smoothing Variable-Bit-Rate Video in an Internetwork", *IEEE/ACM Transactions on Networking*, Vol 7, N° 2, April 1999, pp. 202-215.
- [Roberts93] J.W.Robert, B.Bensaou, and Y.Canetti, "A traffic control framework for high speed data transmission", in *Proc. IFIP Workshop Modeling, Performance Evaluation. ATM Technol.*, 1993, pp. 243-262.
- [Rose95a] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems", *Proceedings of the 20<sup>th</sup> Annual Conference on Local Computer Networks, Minneapolis, MN, 1995*, pp. 397-406. Mpeg traces in: ftp-info3.infomartik.uni-wuerzburg.de in /pub/MPEG/
- [Rose95b] O. Rose, "Simple and Efficient Models for Variable Bit Rate MPEG Video Traffic", Research Report n° 120. Institute of Computer Science, Univesity of Würzburg. July 1995.
- [Ross96] F.E.Ross, D.R. Vaman, "IsoEthernet: An integrated Services LAN", *IEEE Communications Magazine* , August 1996 Pág. 74-84.
- [Salehi98] J.D. Salehi, Z.L.Zhang, J.Kurose and D.Towsley, "Supporting Stored Vieo: Reducing Rate Variability and End-To-End Resource Requirements Through optimal Smoothing", *IEEE/ACM Transactions on Networking*, pp. 397-410.Vol.6, N° 4, August 1998
- [Schenker96] S. Schenker, C.Partridge, R.Guerin "Specification of Guaranteed Quality of Service", *RFC 2212*. 1996.
- [Stuttgen95] H.J.Stuttgen, "Network Evolution and Multimedia Communication" *IEEE Multimedia* . Fall 1995 Pág. 42- 59
- [Tanenbaum96] Andrew S.Tanenbaum, *Computer Networks* .Third Edition. Prentice Hall 1996.
- [Turner86] J.S. Turner, "New directions in communications (or Which Way to the Information age)", *IEEE Communications Magazine*, vol. 24, pp. 8-15, Oct. 1986.
- [USCISI96] ReSerVation Protocol 2 (RSVP2) University of Southern California / Information Sciences Institute. [http://www.ito.darpa.mil/Summaries95/D016--USC\\_ISI\\_ReSerVation2.html](http://www.ito.darpa.mil/Summaries95/D016--USC_ISI_ReSerVation2.html) , 12-Sep-96.
- [Varma97] A. Varma, D. Stialidis "Hardware Implementation of Fair Queuing Algorithms for Asynchronous Transfer Mode Networks" *IEEE Communications Magazine*, December 1997 pp. 54-68.
- [Verma91] D. Verma, H. Zhang, and D.Ferrari, "Guaranteeing Delay Jitter Bounds in Packet Switching Networks", in *Proc. Tricomm '91*.
- [Vogel95] A. Vogel, B. Kerhervé, G. Bochmann and J. Gecsei, "Distributed multimedia and QOS: A survey" , *IEEE Multimedia* Summer 1995, pp. 10-18
- [Wallace91] G.K. Wallace, "The JPEG Still Picture Compression Standard" *Communications of ACM*, Vol. 34, N° 4, Apr. 1991, pp. 30-44.

- [White97] P.P White, "RSVP and Integrated Services in the Internet : A tutorial", *IEEE Communications Magazine*, May 1997 pp. 100-106.
- [Widyono94] R. Widyono, "The Design and Evaluation of Routing Algorithms for Real-Time Channels", *M.S. Report, University of California, Berkeley, CA*, May 1994.
- [Wrege96] E. Wrege, E. W. Knightly, H. Zhang and J. Liebeherr, "Deterministic Delay Bounds for VBR Video in Packet-Switching Networks : Fundamental Limits and Practical Tradeoffs". *IEEE/ACM Transactions on Networking*, 4(3):352-362, June 1996.
- [Wroclawski00] J. Wroclawski and A.Charny. "Integrated Service Mappings for Differentiated Services Networks. Networks - IETF draft, Mar. 2000. <http://www.ietf.org/internet-drafts/draft-ietf-iss11-ds-map-00.txt>
- [Wroclawski96] J.Wroclawski, "Specification of the controlled-Load network Element Service". *RFC 2211*. 1996.
- [Zhang93] H. Zhang, "Service Disciplines for integrated services packet-switching networks". PhD Dissertation. UCB/CSD-94-788. University of California at Berkeley. November 1993.
- [Zhang94] H. Zhang and D. Ferrari, "Rate-Controlled Service Disciplines", to appear in *Journal of High Speed Networks*, accepted in February, 1994.
- [Zhang98] H.Zhang "Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks", *Proceedings of the IEEE*, Vol. 83, N° 10, 1374-1396
- [ZhangL90] L. Zhang, "Virtual Clock : A new traffic Control Algorithm for Packet Switching Networks", in *Proc. ACM SIGCOMM '90*, pp. 19-29.
- [ZhangL93] L. Zhang, S.Deering, D. Estrin, S. Shenker, D. Zappala. "RSVP : A New Resource Reservation Protocol". *IEEE Network Magazine* September 1993.
- [ZhangZ95] Z. L. Zhang, D. Towsley and J. Kurose. "Statistical Analysis of the Generalized Processor Sharing Scheduling Discipline". *IEEE Journal of Selected Areas in Telecommunications*, Vol. 14, N° 6, pp. 1071-1080, Aug. 1995.
- [Zheng92] Q.Zheng and K. G. Shin "Fault-Tolerant Real-Time Communication in Distributed Computing Systems", *Proceedings of the IEEE FTCS*, pp. 86-93, 1992.