



Universidad Politécnica de Cataluña
Programa de Doctorado de Ingeniería Telemática



TESIS DOCTORAL

***Contribución al soporte de
Calidad del Servicio en Redes
Móviles***

Autor: Jhon Jairo Padilla Aguilar

Director: Josep Paradells Aspas

Año: 2007

*A Daniela y María Alejandra, el regalo
más hermoso que Dios me ha dado*

Agradecimientos

Realmente hacer un Doctorado es una tarea muy compleja. Cuando inicié esta fase de mi vida, no tenía idea de lo que realmente era. Ahora, viendo hacia atrás me doy cuenta que para alcanzar esta meta realmente han tenido que darse muchas situaciones y han participado muchas personas en esta empresa. Intentaré incluir a todas, sin embargo, si alguna se me pasa, presento mis disculpas...

En primer lugar quiero dar gracias a Dios, por haberme permitido tener esta experiencia tan enriquecedora para mi vida, en un país y un continente diferente y conocer personas de otras regiones, culturas y formas de ver la vida.

Un agradecimiento muy especial a mi Director de Tesis, Josep Paradells, quien también considero mi amigo, y quien me dio siempre su consejo y me guió por este camino en el aprender a hacer una Tesis Doctoral. También un agradecimiento muy grande a la Universidad Pontificia Bolivariana, especialmente a todas aquellas personas que me dieron su apoyo mediante la Comisión de Estudios para realizar el Doctorado. Ellos son, el Dr. Edgar Barrios, el Dr. Luis Enrique Aramburo, el Dr. Luis Alfonso Díaz, el Dr. Luis Felipe Casas, el Ing. Alex Monclou, el Ing. Samuel Montero y por supuesto, el rector, Monseñor Nestor Navarro Barrera.

Durante mi estancia en Barcelona, he encontrado muchos amigos que han sido mi familia en estas tierras lejanas de mi país natal. Quienes han sido mis compañeros de Piso en diferentes ocasiones: José Carlos Flores, Alejandro Hurtado, Manuel Gallego, María Fernanda Quiroz, Oscar Calderón, Oscar Díaz, Carlos Gaviria, Miller Salas. También a mis compañeros del despacho patera y de la UPC: Mónica Huerta, Oscar Calderón, Patricia Carbajo, Isabel Cristina Satizábal, Rafael Páez, Guillermo Díaz, Rafael Martínez, Miller Salas, Roger Clotet. También otros amigos que han estado presentes en Barcelona o en España han sido Nelson Molina, Diana Agudelo, Luis Angel, Guillermo Bedoya y su esposa Teresa.

En mi estancia en Colombia, también he contado con amigos que han sido mi familia. El grupo del Departamento de Bienestar Universitario: Luz Adriana Gómez, Janed Ardila, Olga Salazar, Yaneth Gonzáles, Doña Libia, Jairo Peña, Luis Gustavo Mateus. También está el grupo de la Facultad de Electrónica: Alex Monclou, Edgar Barrios, Héctor Pérez, Claudia Rueda, Fabio Guzmán y su esposa Liliana Urbano, Jesús Vega, Sayra Cristancho, Dóris Sánchez, Manuel Guillermo Quijano, Martha Janeth Sierra. Y qué decir de mis estancias en Bogotá para sacar el visado para diferentes países, muchas gracias a Diego Rosero y su novia Gina, por darme posada infinidad de veces. Un agradecimiento especial a Paola Durán, Melissa Fandiño, Myriam Victoria Mutis, Claudia Gonzalez, grandes amigas.

Finalmente, un agradecimiento muy grande a mi familia; Lola y Carlos, mis padres; Daniela y Maria Alejandra, mis hijas; Vicky y Natalia, mis hermanas; Juan Manuel, mi sobrino; Leonor, mi abuela; Magdalena Gámez y Jesús Arias, los abuelos maternos de mis hijas y a la madre de mis hijas, Carla. Todos ellos han hecho un gran esfuerzo por apoyarme de una u otra forma, algunos me han tenido paciencia y otros me han dado ánimo cuando desfallecía.

Resumen

En esta Tesis se plantea una solución para soportar *Calidad de Servicio* en *Internet*. Esta solución, denominada *IntServ6*, opera sobre redes fijas y móviles basadas en el protocolo *IPv6*. *IntServ6* está fundamentada en la *Arquitectura de Servicios Integrados* y utiliza el campo de *Etiqueta de Flujo* de la cabecera *IPv6* para identificar los flujos que tienen reservas de recursos en los *Encaminadores*. La utilización del campo *Etiqueta de Flujo* de *IPv6* no ha sido estandarizada, existiendo hasta el momento sólo ciertas especificaciones de uso y algunas propuestas de su utilización que también fueron estudiadas. Tales propuestas sólo abarcan algunos aspectos de una posible solución pero no plantean soluciones completas. Estos aspectos son entre otros, el formato de interpretación de la *Etiqueta de Flujo* y algunas definiciones de arquitecturas de *QoS* descritas de forma muy general. Adicionalmente, *IntServ6* posee extensiones para operar sobre redes móviles. Tales extensiones están diseñadas para operar con los protocolos de movilidad existentes de forma que no requieren la alteración de dichos protocolos. Aunque *IntServ6* podría operar con el protocolo *Mobile IPv6*, es preferible su operación sobre *HMIPv6* para reducir el tiempo de *Traspaso*.

Para la evaluación de *IntServ6*, se obtuvieron diferentes modelos matemáticos y de simulación que permitieron evaluar el rendimiento de la propuesta *IntServ6* con respecto al estándar *IntServ* y, en algunos aspectos, se comparó también con *MPLS*. Todos los aspectos evaluados arrojaron que definitivamente *IntServ6* es superior a *IntServ*. Así, *IntServ6* mejora el rendimiento de los *Encaminadores* en aspectos como el *Ancho de Banda de Memoria* y la velocidad de procesamiento, también mejora el *Tiempo Medio de Retardo* de los paquetes, mejora el *Tráfico Servido* a cada flujo en *Reservas Simples* y en las *Reservas en Túneles*. Por otro lado, *IntServ6* reduce considerablemente la dependencia del *Retardo* de los paquetes con respecto a la movilidad de los usuarios y además no afecta e incluso reduce un poco el *Tiempo de Latencia de Traspaso*.

Abstract

This Thesis has obtained an approach to support *Quality of Service on Internet*. This approach, named *IntServ6*, operates over wired and wireless *All IP Networks* based on the *IPv6* protocol. *IntServ6* is based on the *Integrated Services Architecture* and it identifies traffic flows (with reservations at the routers), by means of the *Flow Label Field* allocated in the *IPv6 Header*. Currently, the use of the *Flow Label Field* is not standardized; thus, only there exist some general specifications about the use of this field and some proposals about it. However, such proposals only describe some aspects (*Flow Label Field* Format and general network architectures) but they not take into account many other issues that *Internet* requires today. In addition, *IntServ6* has been designed with extensions to operate over *Mobile Networks*. Such extensions allow to *IntServ6* to obtain a good performance with the current *Mobility Protocols* without make any modification to them. Although *IntServ6* could operate with the *Mobile IP* protocol, is preferred that it operates with the *Hierarchical Mobile IP* protocol to reduce the *Handoff Latency*.

In order to evaluate *IntServ6*, several mathematical and simulation models were developed. Such models allowed evaluating the performance of *IntServ6* and to compare it with the performance of the standard *IntServ*. In some cases, also *IntServ6* was compared with *MPLS*. All evaluated issues show that *IntServ6* is better than *IntServ*. Thus, *IntServ6* improves the router performance in aspects as the *Memory Bandwidth* and the *Processing Rate*. *IntServ6* reduces the *mean packet delay*; it also increases the *Served Traffic* for each flow in *Simple Reservations* and also in *Reservations within Tunnels*. On the other hand, *IntServ6* reduces the *packet delay* dependence with the user mobility. Finally, *IntServ6* also reduces the *Handoff Latency*.

Índice

1	INTRODUCCIÓN	1
1.1	<i>Motivación</i>	2
1.2	<i>Objetivos de la Tesis</i>	4
1.3	<i>Aportaciones Originales de la Tesis Doctoral</i>	6
1.4	<i>Estructura de la Tesis Doctoral</i>	7
2	ESTADO DEL ARTE	9
2.1	El Protocolo IPv6	9
2.1.1	<i>Motivación de IPv6</i>	9
2.1.2	<i>Estructura de los Paquetes IPv6</i>	10
2.1.3	<i>Cabecera IPv6</i>	13
2.2	Calidad de Servicio	14
2.3	Tecnologías para el soporte de QoS	16
2.3.1	<i>Arquitectura de servicios Integrados</i>	16
2.3.1.1	<i>Modelos de Servicio en IntServ</i>	18
2.3.1.1.1	<i>Especificación de flujos</i>	19
2.3.1.1.2	<i>Servicio garantizado</i>	21
2.3.1.1.3	<i>Servicio de carga controlada</i>	22
2.3.2	<i>Arquitectura de Servicios Diferenciados</i>	23
2.3.2.1	<i>Aproximación Básica</i>	23
2.3.2.2	<i>Estructura del campo DS</i>	25
2.3.2.3	<i>Gestión de Recursos Extremo a Extremo</i>	26
2.3.3	<i>Multi Protocol Label Switching (MPLS)</i>	27
2.4	Procesos para el soporte de QoS	30
2.4.1	<i>Establecimiento de Reservas: El protocolo RSVP</i>	30
2.4.2	<i>Clasificación de flujos</i>	34
2.4.2.1	<i>Definición y tipos de clasificación</i>	34
2.4.2.2	<i>Tagging vs. Matching</i>	35
2.4.3	<i>Planificación de Paquetes</i>	36
2.4.3.1	<i>Requisitos Básicos de un Planificador</i>	37
2.4.3.2	<i>Algoritmos de Planificación tipo Fair-Queuing</i>	38
2.5	Movilidad en redes IP	41
2.5.1	<i>Solución de Macro-Movilidad: El protocolo IP Móvil</i>	42
2.5.1.1	<i>El protocolo Mobile IPv4</i>	42
2.5.1.2	<i>El protocolo Mobile IPv6</i>	46
2.5.1.2.1	<i>Mensajes de Mobile IPv6</i>	46
2.5.1.2.2	<i>Estructuras de datos usadas en MIPv6</i>	46
2.5.1.2.3	<i>Operación de Mobile IPv6</i>	47
2.5.2	<i>Soluciones de Micro-Movilidad</i>	50
2.5.2.1	<i>El protocolo Hierarchical Mobile IP</i>	50
2.5.2.2	<i>NETLMM</i>	51
2.6	QoS en redes móviles todo IP	52
2.6.1	<i>Los problemas a solucionar</i>	53

2.6.2	<i>Soluciones de Calidad del Servicio con IntServ</i>	55
2.6.3	<i>Soluciones de Calidad de Servicio con DiffServ</i>	57
2.7	<i>Alteraciones en el tratamiento de los paquetes IP en diferentes tipos de trayectos</i>	59
2.7.1	<i>Encaminamiento con prefijos y Agregación de flujos en IPv6</i>	60
2.7.1.1	<i>Encaminamiento con prefijos</i>	60
2.7.1.2	<i>Agregación de flujos</i>	61
2.7.2	<i>Tunelización de paquetes genérica en IPv6</i>	63
3	<i>INTSERV6: PROPUESTA PARA EL SOPORTE DE CALIDAD DE SERVICIO EN INTERNET</i>	67
3.1	<i>El vacío en el conocimiento: Uso de la Etiqueta de Flujo de IPv6 para el soporte de QoS.</i>	68
3.2	<i>Clasificación de flujos en IntServ6</i>	74
3.2.1	<i>Clasificación de paquetes con IntServ</i>	74
3.2.2	<i>Clasificación con IntServ6</i>	76
3.3	<i>Reservas dentro de los túneles</i>	78
3.3.1	<i>Reservas dentro de túneles con IntServ</i>	79
3.3.2	<i>Reservas dentro de túneles con IntServ6</i>	80
3.4	<i>Agregación de flujos</i>	83
3.4.1	<i>Agregación de flujos con el estándar actual</i>	84
3.4.2	<i>Agregación de flujos con IntServ6</i>	86
3.5	<i>Interconexión con redes de transporte MPLS</i>	87
3.6	<i>Extensiones para el soporte a la movilidad</i>	90
3.7	<i>Cumplimiento de las especificaciones de la Etiqueta de Flujo en la propuesta IntServ6</i>	95
3.8	<i>Conclusiones</i>	97
4	<i>EVALUACIÓN DE INTSERV6</i>	99
4.1	<i>El procesador de red IXP 1200 de Intel como plataforma de Evaluación</i>	99
4.1.1	<i>El procesador IXP1200</i>	100
4.1.2	<i>Arquitectura de Software del IXP1200</i>	102
4.2	<i>Mejora en el Rendimiento de los Encaminadores</i>	103
4.2.1	<i>Modelo de Hardekopf</i>	104
4.2.2	<i>Mejoras al Modelo de Hardekopf</i>	107
4.2.3	<i>Modelo de IntServ6</i>	108
4.2.4	<i>Resultados Obtenidos</i>	109
4.2.4.1	<i>Relación Ancho de banda de memoria (BW_m)- lecturas tabla hash (k)</i>	110
4.2.4.2	<i>Relación Ancho de Banda de memoria (BW_m)- Número de Flujos</i>	111
4.3	<i>Evaluación del Retardo Medio de los paquetes</i>	114
4.3.1	<i>Metodología de Evaluación para el Retardo medio</i>	114
4.3.2	<i>Desarrollo del modelo matemático del Encaminador</i>	116
4.3.2.1	<i>Modelo de un Encaminador de Servicios Integrados</i>	116
4.3.2.2	<i>Modelo Estocástico del proceso de retransmisión de paquetes</i>	117
4.3.2.3	<i>Justificación del uso de llegadas exponenciales para los modelos propuestos</i>	120
4.3.2.4	<i>Tiempo medio de Servicio de los paquetes en el proceso de clasificación en IntServ6</i>	124
4.3.2.5	<i>Tiempo medio de Servicio de los paquetes en el proceso de clasificación en IntServ</i>	130
4.3.2.6	<i>Modelo de colas del proceso de clasificación</i>	133

4.3.2.7	<i>Modelo del planificador</i>	137
4.3.2.8	<i>Modelo Completo del Encaminador</i>	139
4.3.3	<i>Simulación del proceso real del Clasificador</i>	141
4.3.3.1	<i>Descripción del Modelo de Simulación</i>	141
4.3.3.2	<i>Simulaciones y Resultados Obtenidos</i>	143
4.3.3.2.1	<i>Simulaciones del tiempo de servicio</i>	143
4.3.3.2.2	<i>Simulaciones del retardo medio</i>	149
4.4	<i>Evaluación de las reservas para IntServ6</i>	151
4.4.1	<i>Efecto del tiempo de clasificación en el tráfico servido</i>	153
4.4.1.1	<i>El modelo de Zhang</i>	154
4.4.1.2	<i>Efecto de la clasificación en el tráfico servido</i>	156
4.4.1.3	<i>Variación del tráfico servido en caso de reservas simples</i>	158
4.4.1.4	<i>Variación del tráfico servido en caso de reservas en túneles</i>	159
4.4.2	<i>Resultados Obtenidos</i>	162
4.5	<i>Dependencia entre el tiempo de retardo y la movilidad de los usuarios</i>	165
4.5.1	<i>Antecedentes</i>	165
4.5.2	<i>Modelo del Encaminador de Acceso inalámbrico</i>	166
4.5.3	<i>Modelo de Movilidad y esquemas de Control de Admisión</i>	168
4.5.3.1	<i>Esquema de Limitación de nuevas llamadas (New Call Bounding Scheme)</i>	168
4.5.3.2	<i>Esquema de Prioridad de Limitación (Cutoff Priority Scheme)</i>	169
4.5.3.3	<i>Número de flujos en el Encaminador de Acceso Inalámbrico</i>	170
4.5.3.4	<i>Retardo medio de los paquetes para el proceso de Clasificación y el proceso de Planificación</i>	171
4.5.4	<i>Resultados Obtenidos</i>	171
4.6	<i>Evaluación del tiempo de Traspaso</i>	174
4.6.1	<i>Arquitectura de red a utilizar</i>	174
4.6.2	<i>Análisis del Ahorro en tiempo de Traspaso para IntServ6</i>	175
4.7	<i>Comparación de IntServ6 con MPLS en redes móviles basadas en IP</i>	179
4.7.1	<i>Introducción</i>	179
4.7.2	<i>Soluciones de MIP/MPLS</i>	180
4.7.3	<i>La gestión de Etiquetas en MIP/MPLS</i>	182
4.7.4	<i>Desempeño de la gestión de etiquetas en MIP/MPLS</i>	183
4.7.5	<i>Comparación de MPLS con IntServ6 en redes de acceso MIP</i>	189
4.8	<i>Conclusiones</i>	190
5	CONCLUSIONES Y LÍNEAS FUTURAS	193
6	LISTA DE ACRÓNIMOS	197
7	PUBLICACIONES RELACIONADAS CON LA TESIS DOCTORAL	199
8	BIBLIOGRAFÍA	201

Listado de Figuras

Figura 2.1. Formato general de un paquete <i>IPv6</i>	11
Figura 2.2 Paquete <i>IPv6</i> con las cabeceras de extensión (conteniendo un segmento <i>TCP</i>).	12
Figura 2.3 Cabecera <i>IPv6</i>	14
Figura 2.4 Modelo de referencia de <i>IntServ</i>	17
Figura 2.5 Operación del <i>Leaky Bucket</i>	20
Figura 2.6 Arquitectura de una Red <i>DiffServ</i>	24
Figura 2.7 Campo <i>IP TOS</i>	25
Figura 2.8 Campo <i>DS</i>	26
Figura 2.9 Arquitectura de una red <i>MPLS</i>	29
Figura 2.10 Establecimiento de reservas con el protocolo <i>RSVP</i>	32
Figura 2.11 Formato de los mensajes <i>RSVP</i>	32
Figura 2.12 Formato de la cabecera Común <i>RSVP</i> y de la cabecera de objeto <i>RSVP</i>	33
Figura 2.13 Formato del objeto <i>SenderTspec</i>	34
Figura 2.14 Entidades de <i>Mobile IPv4</i>	43
Figura 2.15 Escenario <i>MIPv6</i>	48
Figura 2.16 Arquitectura usada en <i>HMIPv6</i>	50
Figura 2.17. Red <i>NETLMM</i>	52
Figura 2.18 Formato de dirección <i>Unicast</i> global agregable de <i>IPv6</i>	61
Figura 2.19 Ejemplo de Encaminamiento con prefijo.	63
Figura 2.20 Tipos de túneles en <i>IPv6 Tunneling</i>	64
Figura 2.21 Encapsulado <i>IPv6</i>	64
Figura 2.22 Des-encapsulado <i>IPv6</i>	65
Figura 2.23 Encapsulación anidada	66
Figura 3.1 Operación del esquema de <i>QoS IPv6</i>	73
Figura 3.2 Procedimiento de <i>Hashing</i> en <i>IntServ</i>	75
Figura 3.3 Tablas de manejo de las reservas en <i>IntServ6</i>	77
Figura 3.4 Diagrama de flujo del procesamiento de paquetes durante la clasificación en un <i>Encaminador IntServ6</i>	78
Figura 3.5 Escenario de Reservas dentro de túneles	80
Figura 3.6 Diferenciación de Reservas dentro de túneles con <i>IntServ6</i>	82
Figura 3.7 Agregación de flujos <i>RSVP</i> estándar.	84
Figura 3.8 Agregación de flujos en <i>IntServ6</i>	86
Figura 3.9 Interconexión de una red <i>IntServ6</i> a una red <i>MPLS</i> utilizando Correspondencia de reservas automática.	89
Figura 3.10 Arquitectura de la red <i>IntServ6 Móvil</i>	92
Figura 3.11 <i>Traspaso Local</i> en una red <i>IntServ6 Móvil</i>	93
Figura 3.12 <i>Traspaso Inter-Dominio</i> en una red <i>IntServ6 Móvil</i>	94
Figura 4.1 Diagrama de Bloques del sistema <i>IXP1200</i> [83].	101
Figura 4.2 Situación analizada en el modelo original	106
Figura 4.3 Variación del <i>Ancho de Banda de Memoria</i> con el <i>Número de Lecturas Hash</i>	111
Figura 4.4 Variación <i>Ancho de banda de memoria</i> con respecto al <i>número de flujos</i> . ..	112
Figura 4.5 Variación del <i>Ancho de banda de memoria</i> con respecto al <i>número de flujos</i> (escala: 1000).....	113
Figura 4.6 <i>Throughput</i> del proceso de clasificación del <i>Encaminador</i>	114

Figura 4.7 Modelo de un Encaminador de <i>Servicios Integrados</i>	117
Figura 4.8 Esquema general del proceso de <i>retransmisión</i> de paquetes en un <i>Encaminador de Servicios Integrados</i>	118
Figura 4.9 Modelo Estocástico del <i>Encaminador IntServ6</i>	120
Figura 4.10 Diagrama de transición de estados del proceso de <i>clasificación de paquetes</i> de un <i>Encaminador IntServ6</i>	125
Figura 4.11 Diagrama de Transición de estados del <i>clasificador InServ</i>	131
Figura 4.12 Procesos de la simulación del <i>Clasificador</i>	142
Figura 4.13 Intervalos de Confianza para <i>IntServ</i>	146
Figura 4.14 Intervalos de Confianza para <i>IntServ6</i>	146
Figura 4.15 Simulación del <i>Tiempo Medio de Servicio</i> de <i>IntServ</i> respecto al <i>Número</i> <i>de Flujos</i> en el <i>Encaminador</i>	147
Figura 4.16 Simulación del <i>Tiempo Medio de Servicio</i> de <i>IntServ6</i> respecto al <i>Número</i> <i>de Flujos</i> en el <i>Encaminador</i>	148
Figura 4.17 Comparación de las simulaciones del <i>Tiempo Medio de Clasificación</i> en <i>IntServ</i> e <i>IntServ6</i>	148
Figura 4.18 Simulación del <i>Retardo</i> de los paquetes en el <i>Clasificador</i>	149
Figura 4.19 Resultados del <i>Retardo</i> de los paquetes para el modelo teórico.	150
Figura 4.20 Comparación del <i>Retardo Medio</i> de paquete en ambas tecnologías para menos de 100 flujos.....	151
Figura 4.21 Comportamiento del <i>tráfico de llegada</i> y el <i>tráfico servido</i> para el flujo <i>i-</i> <i>ésimo</i> en un planificador <i>GPS</i>	154
Figura 4.22 Diagrama de estados de un <i>Encaminador IntServ</i> para procesar reservas dentro de un túnel	160
Figura 4.23 Ganancia en el <i>límite superior del tráfico servido</i> cuando se varía el <i>Número de Flujos</i> en reservas simples.	162
Figura 4.24 Ganancia de <i>IntServ6</i> con respecto a <i>IntServ</i> en el <i>límite superior de tráfico</i> <i>servido</i>	163
Figura 4.25 Ganancia de <i>IntServ</i> con respecto a <i>IntServ6</i> en el límite superior de tráfico servido.	164
Figura 4.26 Modelo de un <i>Encaminador de Acceso Inalámbrico</i>	167
Figura 4.27 Variación del <i>Retardo Medio</i> de las ráfagas de paquetes para <i>IntServ</i> e <i>IntServ6</i> , bajo dos esquemas de <i>Control de Admisión</i> , con respecto a la <i>variación de la</i> <i>intensidad de tráfico de Nuevas Llamadas</i>	172
Figura 4.28 Variación del <i>Retardo Medio</i> de las ráfagas de paquetes para <i>IntServ</i> e <i>IntServ6</i> , bajo dos esquemas de <i>Control de Admisión</i> , con respecto a la <i>variación de la</i> <i>intensidad de tráfico</i> de llamadas de <i>Traspaso</i>	173
Figura 4.29 Arquitectura usada en <i>HMIPv6</i>	175
Figura 4.30 Fase de establecimiento de la reserva.	176
Figura 4.31 Fase de <i>Fast Handover</i> entre ARs de un mismo dominio <i>MAP</i>	176
Figura 4.32 Fase de <i>Traspaso</i> entre <i>MAPs</i>	177
Figura 4.33. Arquitectura de la red simulada	184
Figura 4.34 Porcentaje de paquetes entregados con respecto al número de etiquetas dinámicas.[114]	185
Figura 4.35 Porcentaje de paquetes entregados con respecto al número de Nodos Móviles[114]	186
Figura 4.36 Consumo de etiquetas en cada <i>LSR</i> respecto al número de etiquetas asignadas [114].	187

Figura 4.37 Consumo de etiquetas en cada <i>LSR</i> respecto al número de móviles [114].	188
Figura 4.38 Porcentaje de paquetes transportados por <i>MPLS</i> respecto al número de etiquetas.....	188

Listado de Tablas

Tabla 2.1	Definición del campo <i>IP TOS</i>	25
Tabla 2.2	Tipos de mensajes <i>RSVP</i>	33
Tabla 3.1	Cumplimiento de las especificaciones de las <i>Etiquetas de Flujo IPv6</i>	95
Tabla 4.1	Características de las arquitecturas en el modelo general	110
Tabla 4.2	Diferencias entre InServ e <i>IntServ6</i>	130
Tabla 4.3	Parámetros de Simulación del procesador de red.....	143
Tabla 4.4	Método de los momentos para <i>IntServ</i>	145
Tabla 4.5	Método de los momentos para <i>IntServ6</i>	145
Tabla 4.6	Valores de tiempo medio de clasificación de las reservas en túneles	164

1 Introducción

El objetivo de esta tesis ha sido realizar aportaciones al soporte de *Calidad de Servicio* en Redes móviles. Es importante anotar que en la actualidad estamos en un momento de cambio de tecnologías de conmutación en las redes móviles. Hasta ahora, se ha manejado la información de estas redes mediante tecnologías de conmutación de circuitos. Sin embargo, las nuevas redes de tercera generación para telefonía celular están cambiando esto para llevar este tipo de redes hacia la conmutación de paquetes. Por otro lado, han surgido tecnologías tales como *Wi-Fi* y *WiMAX* que permiten el acceso a *Internet* de una forma inalámbrica. Debido a esta situación de transición, los métodos de análisis de tráfico y diseño de estas nuevas redes móviles están en una etapa inicial de estudio, mientras que los métodos de análisis y diseño de las tecnologías de conmutación de circuitos están bastante decantados y presentan una gran madurez.

La llegada de las tecnologías de conmutación de paquetes hace que la tendencia de las redes de próxima generación sea que estas deban soportar el protocolo *IP* para el transporte de los paquetes. Así, estas redes de próxima generación serán redes móviles basadas en *IP*, por lo que el escenario de trabajo de esta tesis se ha centrado en este tipo de redes. Las redes móviles basadas en *IP* deben solucionar adicionalmente el problema de la movilidad de los usuarios en una *Internet* que fue diseñada inicialmente para redes fijas, por lo que utilizan una solución ya estandarizada para este problema y que se conoce como el protocolo *IP Móvil*. Dicho protocolo ha sido desarrollado en dos versiones, una para el protocolo *IPv4* y otra para el protocolo *IPv6*. Una modificación de este protocolo, conocida como *IP Móvil Jerárquico (Hierarchical Mobile IP)*, fue desarrollada para disminuir la cantidad de mensajes de señalización en entornos en los que se cambia muy frecuentemente de un *Encaminador* de acceso inalámbrico a otro, y organiza la red de una forma jerárquica. Igualmente existen dos versiones de esta solución, una para *IPv4* y otra para *IPv6*.

Uno de los principales aspectos que deben atacarse en las nuevas redes móviles de conmutación de paquetes basadas en *IP* es brindar soporte a la *Calidad de Servicio* para los usuarios. Esto quiere decir, que estas redes deben permitir diferentes niveles de

servicio para diferentes usuarios. Estos niveles de servicio incluyen el soporte de aplicaciones que van desde aplicaciones elásticas, tales como el *Web* y la transferencia de archivos, hasta aplicaciones de tiempo real como la transmisión de voz y vídeo.

Esta tesis propone una solución para soportar *Calidad de Servicio en Internet*. Esta propuesta se conoce como *IntServ6* y puede aplicarse tanto a una *Internet* fija como a una *Internet* móvil. *IntServ6* está basada en la *Arquitectura de Servicios Integrados* pero, a diferencia de esta, *IntServ6* explota las características de *IPv6* que permiten soportar *Calidad de Servicio*, aspecto que ha sido previsto en los campos de la cabecera de *IPv6* mediante el campo etiqueta de flujo, pero que en la práctica no ha sido utilizado y mucho menos estandarizado. Los cambios realizados a la arquitectura *IntServ* estándar para conformar la propuesta *IntServ6* incluyen una serie de mejoras en aspectos problemáticos tales como el mejoramiento del desempeño de los *Encaminadores*, la disminución del problema de *Escalabilidad*, el establecimiento de *Reservas en Túneles* y la *Agregación de Flujos*.

Este proyecto de tesis se encuentra enmarcado dentro del proyecto *RUBI (Red Ubicua Basada en IP)* apoyado por el *CICYT* (código: *TIC 2003-01748*) bajo la responsabilidad del *WNG (Wireless Network Group)* del departamento de Ingeniería Telemática de la *UPC*.

Finalmente, debido a que en las nuevas tecnologías aparecen palabras que no han sido aún definidas en el Diccionario de la Real Academia de la Lengua Española, y con fines de mantener una terminología estándar para el idioma español, se ha utilizado en este documento la terminología de la *UIT* (Unión Internacional de Telecomunicaciones)¹.

1.1 Motivación

Las motivaciones por las cuales se ha desarrollado esta tesis doctoral son las siguientes:

- Diseñar una propuesta para el soporte de *Calidad de Servicio en Internet* basada en el concepto de flujos:

¹ La terminología de la *UIT* se puede consultar en la dirección <http://www.itu.int/sancho/index.asp>

La meta es plantear soluciones a nivel de la capa *IP* (sacando provecho a las características del protocolo *IPv6*), suponiendo que problemas de la capa de acceso a la red ya han sido solucionados o serán solucionados prontamente por los fabricantes. Uno de los principales supuestos es que los ordenadores móviles deben tener el *Hardware* necesario para pasar de una red de acceso a otra fácilmente, es decir, tendrán tarjetas de red que soporten los diferentes tipos de tecnologías de la capa de acceso.

- La propuesta de *Calidad de Servicio* debe trabajar tanto en entornos fijos como en entornos móviles.

La llegada de las tecnologías de conmutación de paquetes a las redes móviles hace que la tendencia de las redes de próxima generación sea que estas deban soportar el protocolo *IP* para el transporte de los paquetes. Así, estas redes de próxima generación tendrán una mezcla de redes fijas y redes móviles basadas en *IP*, por lo que el escenario de trabajo de esta tesis se ha centrado en este tipo de redes. Las redes móviles basadas en *IP* deben solucionar adicionalmente el problema de la movilidad de los usuarios en una *Internet* que fue diseñada inicialmente para redes fijas, por lo que utilizan una solución ya estandarizada para este problema y que se conoce como el protocolo *IP Móvil*. Dicho protocolo ha sido desarrollado en dos versiones, una para el protocolo *IPv4* y otra para el protocolo *IPv6*. Una modificación de este protocolo, conocida como *IP Móvil Jerárquico (Hierarchical Mobile IP)*, fue desarrollada para disminuir la cantidad de mensajes de señalización en entornos en los que se cambia muy frecuentemente de un *Encaminador* de acceso inalámbrico a otro, y organiza la red de una forma jerárquica. Igualmente existen dos versiones de esta solución, una para *IPv4* y otra para *IPv6*.

Uno de los principales aspectos que deben atacarse en las nuevas redes móviles de conmutación de paquetes basadas en *IP* es brindar soporte a la *Calidad de Servicio* para los usuarios. Esto quiere decir, que estas redes deben permitir diferentes niveles de servicio para diferentes usuarios. Estos niveles de servicio incluyen el soporte de aplicaciones que van desde aplicaciones elásticas, tales

como el *Web* y la transferencia de archivos, hasta aplicaciones de tiempo real como la transmisión de voz y vídeo.

1.2 Objetivos de la Tesis

El objetivo de esta tesis es hacer contribuciones al soporte de la *Calidad de Servicio* en redes móviles basadas en *IP*. Para ello se debieron cumplir objetivos específicos como:

1. Estudio de la problemática actual de *Calidad de Servicio* en redes móviles
2. Proponer una arquitectura de red móvil basada en *IP*
3. Desarrollo de una propuesta de *Calidad del Servicio* en redes móviles
4. Validar la propuesta de *Calidad de Servicio*

A continuación se detallan las tareas desarrolladas en la tesis para cumplir con los objetivos planteados:

1. Estudio del estado del arte en movilidad:

Se estudiaron los problemas de macro-movilidad y micro-movilidad y los protocolos que los solucionan, es decir, *Mobile IP* [1, 2] para el caso de macro-movilidad y *Hierarchical Mobile IP* [3], *cellular IP* [4] y *HAWAII* [5] para el caso de micro-movilidad.

2. Estudio del estado del arte en Calidad de Servicio en Redes Móviles IP:

Se estudiaron diversas propuestas existentes en el soporte de *Calidad de Servicio* en redes móviles basadas en *IP*.

3. Estudio de las diferentes tecnologías actuales y las tendencias futuras en redes móviles:

Se estudiaron las diferentes tecnologías actuales de redes móviles tales como *IEEE 802.11*, *GPRS*, *UMTS* y las tendencias futuras hacia las redes *4G*.

4. *Determinación de la arquitectura de red:*

En base a los resultados de las actividades anteriores se propuso una arquitectura de red que permita la movilidad basándose en el protocolo *IP* y que usa diferentes tecnologías de acceso inalámbrico.

5. *Diseño de la Propuesta del modelo de Calidad de Servicio:*

En base a la información recopilada en las anteriores actividades se hizo una propuesta para soportar *Calidad de Servicio* que opera tanto en redes *IP* fijas como en redes móviles basadas en *IP*.

6. *Evaluación de la propuesta de Soporte de QoS*

En esta fase de la tesis se determinaron los diferentes aspectos a evaluar de la propuesta tales como el desempeño de los *Encaminadores*, el retardo medio de los paquetes, la dependencia del retardo de los paquetes con respecto a la movilidad de los usuarios, el tiempo de *Traspaso* y la mejora del ancho de banda tanto en reservas simples como en reservas dentro de túneles.

Para realizar la evaluación de cada uno de estos aspectos se analizó cada uno de los procesos y se desarrollaron los respectivos modelos matemáticos. Estos procesos fueron simulados y se obtuvieron las curvas que muestran el comportamiento de cada uno de ellos. Los resultados obtenidos en las diferentes curvas fueron analizados y se obtuvieron conclusiones sobre cada uno de los aspectos. Los resultados obtenidos, junto con las respectivas simulaciones y modelos matemáticos para cada una de las situaciones a evaluar fueron publicados en diferentes congresos y revistas.

7. *Desarrollo del Informe final y presentación de la Tesis:*

En esta actividad se realizó el informe final de los resultados de la Tesis.

1.3 Aportaciones Originales de la Tesis Doctoral

En esta tesis se obtiene una solución para el soporte de *Calidad de Servicio* sobre redes *IP* fijas y redes móviles basadas en *IP* que mejora diferentes características de la tecnología *IntServ* estándar. Esta propuesta se denomina *IntServ6* y presenta ventajas tales como: la mejora en el desempeño de los *Encaminadores*, la reducción del tiempo de retardo de los paquetes en los *Encaminadores*, la reducción de la dependencia del retardo de los paquetes en los *Encaminadores de Acceso* con respecto a la movilidad de los usuarios, la reducción del tiempo de *Traspaso*, el aumento de la cantidad de tráfico servido en los túneles utilizados en los protocolos de movilidad y el soporte de métodos más eficientes para realizar agregación de flujos e interconexión con redes *MPLS*.

Las aportaciones específicas que hace esta tesis doctoral son:

1. Se obtiene un método eficiente de clasificación de paquetes en *Encaminadores de Servicios Integrados*. Dicho método hace uso del campo *Etiqueta de Flujo* de *IPv6* para la identificación de los flujos y ofrece además una serie de características que hasta el momento no son posibles o no tienen soluciones eficientes en las redes de *Servicios Integrados* que operan con la normativa actual.
2. Se obtiene una solución para el problema del *Soporte de Calidad de Servicio en Túneles* en redes *IPv6* basadas en *Servicios Integrados*.
3. Se obtiene una solución para el problema de la *Agregación de Flujos* en redes de *Servicios Integrados*.
4. Se obtiene un método de Interconexión eficiente entre *Redes de Acceso de Servicios Integrados* y *Redes de Transporte MPLS*.
5. Se obtienen extensiones para el soporte de la *Movilidad de las Reservas* en *Redes Móviles Basadas en IP*.

6. Se obtiene un modelo del desempeño de los *Encaminadores de Servicios Integrados* que permite la evaluación de parámetros tales como el *Número de Hilos de Procesamiento* necesarios y el *Ancho de Banda de Memoria* utilizado durante la retransmisión de los paquetes.
7. Se obtiene un modelo matemático para un *Encaminador de Servicios Integrados* que permite realizar cálculos del *Tiempo Medio de Retardo* de los paquetes durante su retransmisión.
8. Se obtiene un modelo matemático que permite evaluar el efecto de la *Movilidad de los Usuarios* sobre el *Retardo Medio* de los paquetes en un *Encaminador de Acceso Inalámbrico* basado en *Servicios Integrados*.
9. Se obtiene un modelo que permite evaluar la *Ganancia en el Tráfico Servido* en *Reservas Simples* y en *Reservas en Túneles* cuando se utiliza *IntServ6*.
10. Se realiza un análisis del ahorro en el *Tiempo de Traspaso* que se obtiene cuando se utiliza *IntServ6*.

1.4 Estructura de la Tesis Doctoral

La presente Tesis Doctoral está estructurada como se describe a continuación. En el capítulo 2 se hace un breve recuento de las diferentes tecnologías que se utilizan en las *Redes Móviles Basadas en IP*; dichas tecnologías abarcan soluciones para obtener la movilidad de los usuarios que cambian de dirección *IP* al desplazarse de un *Encaminador de Acceso* a otro (protocolos de movilidad); también se describen las arquitecturas estándar para soportar *Calidad de Servicio* en *Internet*. Además, se describen algunas soluciones para el soporte de *Calidad de Servicio* en *Redes Móviles Basadas en IP* y se describen algunos procesos necesarios para el soporte de *Calidad de Servicio* en los *Encaminadores de Servicios Integrados*. Finalmente, se explican algunas situaciones típicas de redes *IP* que alteran el tratamiento de los paquetes tales como la *Agregación de Flujos* y los *Túneles*.

El capítulo 3 describe la propuesta objeto de esta Tesis para el Soporte de *Calidad de Servicio* en *Internet*. Esta propuesta se denomina *IntServ6* y está basada en la *Arquitectura de Servicios Integrados*. Se describen varios aspectos relevantes para el buen funcionamiento de la propuesta en una *Internet* que presenta diversas situaciones tales como el *Soporte de Reservas en Túneles*, *Agregación de Flujos*, *Interconexión con Redes MPLS* y *Movilidad de los Usuarios*. Varias de estas situaciones son descritas para el caso de los estándares actuales para *Servicios Integrados* y luego se describe cómo la propuesta *IntServ6* mejora las deficiencias que presentan los estándares actuales. Los temas tratados en este capítulo fueron publicados en [6].

El capítulo 4 presenta la evaluación de *IntServ6* en diferentes aspectos y su comparación con respecto a la solución estándar *IntServ*. Se estudian aspectos como el desempeño de los *Encaminadores* mediante parámetros como el *Ancho de Banda de Memoria* y el rendimiento con respecto al *Número de Hilos* utilizado durante el procesamiento (publicación en [7-9]). También se estudia el *Retardo Medio de los Paquetes* (publicación en [6]), la dependencia del retardo de los paquetes con respecto a la movilidad de los usuarios en *Encaminadores de Acceso* (publicación en [6, 10, 11]), el *Tiempo de Traspaso* (publicación en [12]), la *Cantidad de Tráfico Servido* en diferentes situaciones tales como *Reservas Simples* y *Reservas Dentro de Túneles* y por último se realiza una comparación con *MPLS* en lo relacionado con la *Gestión de Etiquetas* en *Redes Móviles de Acceso*.

Finalmente, en el capítulo 5 se describen las conclusiones de la investigación realizada y en el capítulo 6 se hacen algunas recomendaciones sobre investigaciones futuras producto de esta tesis (se han hecho algunos trabajos iniciales en las publicaciones [13-15]).

2 Estado del Arte

2.1 El Protocolo IPv6

Debido a que esta Tesis se enfoca en brindar *QoS* en redes *IPv6*, y también debido a que la propuesta planteada utiliza el campo de *Etiqueta de Flujo* de la cabecera *IPv6*, es necesario hacer una breve descripción del formato de dicha cabecera. Además, previamente se hará una breve introducción al protocolo *IPv6*. La descripción encontrada en esta sección puede encontrarse más detallada en [16, 17].

2.1.1 Motivación de IPv6

El motivo principal que ha conducido a la adopción de una nueva versión del protocolo *IP* ha sido la limitación impuesta por el campo de dirección de 32 bits en *IPv4*, de forma que se ha producido un agotamiento de las direcciones *IP* a causa del gran auge del *Internet* pues el número de ordenadores que la utilizan ha crecido enormemente. De otra parte, este agotamiento de las direcciones también es consecuencia de una organización y asignación ineficiente del espacio de direcciones del protocolo *IPv4*, lo que ha ocasionado la infra-utilización de las posibles combinaciones con los bits de direcciones.

En respuesta a estas y otras necesidades actuales, la IETF generó un nuevo estándar denominado *IPv6*, cuya especificación general se encuentra en el *RFC 2460* [18]. Esta nueva versión de *IP* mejoró además otras características de *IPv4*, entre las que se encuentran:

- *Un espacio de direcciones ampliado: IPv6* utiliza direcciones de 128 bits en lugar de las direcciones de 32 bits de *IPv4*.

- *Un mecanismo de Opciones mejorado:* Las opciones de *IPv6* se encuentran en cabeceras opcionales separadas situadas entre la cabecera *IPv6* y la cabecera de la capa de Transporte. La mayoría de estas cabeceras no se examinan ni procesan en ningún dispositivo de encaminamiento en la trayectoria del paquete. Esto simplifica y acelera el procesamiento que realiza un dispositivo de encaminamiento sobre los paquetes *IPv6* en comparación con los *Datagramas* de *IPv4*. Esto también hace que sea más fácil incorporar opciones adicionales.
- *Direcciones de Autoconfiguración:* Esta capacidad proporciona una asignación dinámica de direcciones *IPv6*.
- *Aumento de la Flexibilidad en el Direcccionamiento:* *IPv6* incluye el concepto de una dirección de mono-distribución (*Anycast*), mediante la cual un paquete se entrega sólo a un nodo seleccionado de entre un conjunto de nodos. Se mejora la escalabilidad del encaminamiento multi-distribución (*Multicast*) con la incorporación de un campo de acción en las direcciones multi-distribución.
- *Facilidad para la asignación de recursos:* En lugar del campo *Tipo de Servicio* de *IPv4*, *IPv6* habilita el etiquetado de los paquetes como pertenecientes a un flujo de tráfico particular para el que el emisor solicita un tratamiento especial. Esto ayuda al tratamiento del tráfico especializado como el de vídeo en tiempo real.

2.1.2 Estructura de los Paquetes IPv6

Una unidad de datos del protocolo *IPv6* (conocida como paquete) tiene el formato general que se observa en la Figura 2.1. La única cabecera indispensable es la cabecera *IPv6*; esta tiene una longitud fija de 40 octetos. Sin embargo, como se ha mencionado antes, se pueden agregar otras cabeceras al paquete *IPv6*, denominadas *Cabeceras de Extensión*. Estas pueden ser de los siguientes tipos:

- *Cabecera de opciones salto a salto*: Define opciones especiales que requieren el procesamiento en cada salto.
- *Cabecera de encaminamiento*: Proporciona un encaminamiento ampliado, similar al encaminamiento por la fuente de IPv4.
- *Cabecera de Fragmentación*: Contiene información de fragmentación y reensamblaje.
- *Cabecera de Autenticación*: Proporciona la integridad del paquete y la autenticación.
- *Cabecera de Encapsulado de la carga de seguridad*: Proporciona seguridad.
- *Cabecera de las opciones para el destino*: Contiene información adicional para que sea examinada en el nodo destino.

Cabecera IPv6	Cabecera de Extensión	...	Cabecera de Extensión	PDU del nivel de transporte
---------------	-----------------------	-----	-----------------------	-----------------------------

Figura 2.1. Formato general de un paquete IPv6.

El estándar IPv6 recomienda que, en caso de que se usen varias cabeceras de extensión, las cabeceras IPv6 aparezcan en el siguiente orden:

1. Cabecera IPv6 (Obligatoria, debe aparecer siempre primero).
2. Cabecera de opciones Salto a Salto.
3. Cabecera de las opciones para el destino: Para opciones a procesar por el primer destino que aparece en el campo dirección IPv6 de destino y por los destinos subsiguientes indicados en la cabecera de encaminamiento.
4. Cabecera de encaminamiento.
5. Cabecera de Fragmentación.
6. Cabecera de Autenticación.

7. Cabecera de encapsulado de la carga de seguridad.
8. Cabecera de opciones para el destino: Para opciones a procesar solamente por el destino final del paquete.

La Figura 2.2 muestra un ejemplo de un paquete *IPv6* que incluye un ejemplar de cada cabecera, excepto aquellas relacionadas con la seguridad. Obsérvese que la cabecera *IPv6* y cada cabecera de extensión incluyen el campo de *Cabecera Siguiente*. Este campo identifica el tipo de cabecera que viene a continuación. Si la siguiente cabecera es de extensión, entonces este campo contiene el identificador del tipo de esa cabecera. En caso contrario, este campo contiene el identificador del protocolo de la capa superior que está usando a *IPv6* (normalmente un protocolo de la capa de transporte), utilizando el mismo valor que el campo *Protocolo* de *IPv4*. En la Figura 2.2 el protocolo de la capa superior es *TCP*, por lo tanto, los datos de la capa superior transportados por el paquete *IPv6* constan de una cabecera *TCP* seguido por un bloque de datos de aplicación.

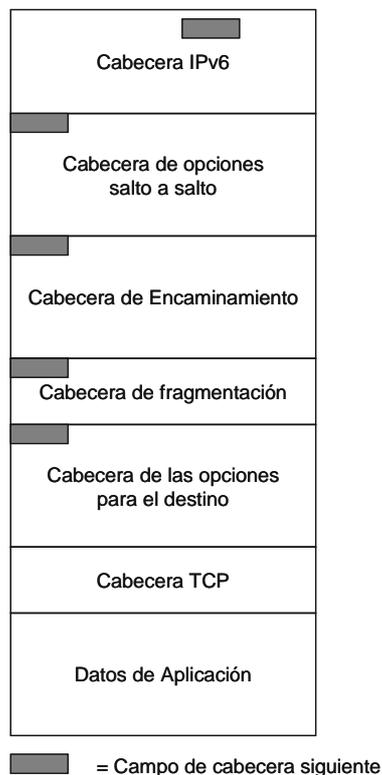


Figura 2.2 Paquete *IPv6* con las cabeceras de extensión (conteniendo un segmento *TCP*).

2.1.3 Cabecera IPv6

La cabecera IPv6 tiene una longitud fija de 40 octetos, que consta de los siguientes campos (Figura 2.3):

- *Versión* (4 bits): Número de la versión del protocolo de Internet; el valor es 6.
- *Clase de Tráfico* (8 bits): Disponible para su uso por el nodo origen o los dispositivos de encaminamiento de re-envío para identificar y distinguir entre diferentes clases o prioridades de paquete IPv6.
- *Etiqueta de Flujo* (20 bits): Se puede utilizar por un ordenador para etiquetar aquellos paquetes para los que se requiere un tratamiento especial en los dispositivos de encaminamiento dentro de la red.
- *Longitud de la carga útil* (16 bits): Longitud en octetos del resto del paquete IPv6, excluida la cabecera IPv6.
- *Cabecera Siguiente* (8 bits): Identifica el tipo de cabecera que sigue inmediatamente a la cabecera IPv6.
- *Límite de Saltos* (8 bits): El número restante de saltos permitidos para este paquete. El límite de saltos se establece por la fuente según el valor máximo deseado; este campo se decrementa en 1 en cada nodo que re-envía el paquete. El paquete se descarta si el límite de saltos se hace cero.
- *Dirección Origen* (128 bits): Dirección del nodo generador del paquete.
- *Dirección Destino* (128 bits): Dirección del destino deseado para el paquete.

Versión	Clase de Tráfico	Etiqueta de Flujo	
Longitud de la carga útil		Cabecera Siguiente	Límite de Saltos
Dirección de Origen			
Dirección de Destino			

Figura 2.3 Cabecera IPv6.

Aunque la cabecera IPv6 es más grande que la parte obligatoria de la cabecera IPv4 (40 octetos frente a 20 Octetos), contiene menos campos (8 frente a 12). Así, los dispositivos de encaminamiento tienen que hacer menos procesamiento por paquete, lo que agiliza el encaminamiento.

2.2 Calidad de Servicio

La Internet actual es una red que ofrece un servicio de tipo *Best-Effort*, esto quiere decir, un servicio en el cual la red tratará de hacer su mejor esfuerzo para entregar los paquetes que recibe, pero la red no se compromete a garantizar la entrega de dichos datos. Estos paquetes podrían no llegar al destino por múltiples razones tales como la congestión, daños en enlaces entre *Encaminadores*, etc. Además, Internet no permite garantizar otros aspectos como el retardo de los paquetes y la reserva de ancho de banda para ciertas aplicaciones. Sin embargo, en la actualidad han surgido nuevas aplicaciones de tiempo real tales como vídeo-conferencia, o la transmisión de voz, las cuales requieren que la red que las soporta les garantice un mínimo de recursos para operar efectivamente. Además, Internet debe soportar diferentes tipos de aplicaciones que tienen diferentes tipos de requisitos.

Los requisitos de servicio son específicos según las aplicaciones. Por ejemplo, un programa de audio sería sensible a un retardo de ida y vuelta; Una aplicación de

transferencia de archivos es más sensible a la tasa de transmisión promedio o a la cantidad total de tiempo necesaria para transferir los datos. Los siguientes parámetros son comúnmente utilizados para describir requisitos de QoS:

- *Ancho de Banda Mínimo*: Es la mínima cantidad del ancho de banda requerida por el flujo de una aplicación. Es necesario especificar el intervalo de tiempo para la medición del ancho de banda ya que diferentes intervalos de medición pueden producir diferentes resultados.
- *Retardo (delay)*: El retardo requerido puede ser especificado como el promedio de los retardos (*Retardo Medio*) o por el retardo del peor caso. El retardo que un paquete experimenta tiene tres componentes: retardo de propagación, retardo de transmisión, y retardo de procesamiento. El retardo de propagación es limitado por la velocidad de la luz, y al mismo tiempo es una función de la distancia. El retardo de transmisión es el tiempo que tarda en enviarse un paquete en un enlace y depende de la longitud del paquete y de la velocidad del enlace; finalmente, el retardo de procesamiento es el tiempo de espera que experimenta un paquete en las colas de los *Encaminadores*.
- *Variación de Retardo (Delay Jitter)*: Este parámetro especifica la máxima diferencia entre el más largo y el más corto retardo que un paquete experimenta. En cualquier caso, la variación de retardo no debería ser más larga que el retardo del peor caso ni tampoco que el retardo de procesamiento.
- *Tasa de Pérdidas (Loss Rate)*: Es el cociente resultante entre los paquetes perdidos y el total de los paquetes transmitidos. La pérdida de paquetes en una *Internet* se debe usualmente a la congestión, y tales pérdidas pueden ser prevenidas mediante la asignación de suficiente ancho de banda y suficiente almacenamiento intermedio (*Buffers*) para el flujo de tráfico.

A la capacidad de una red para asegurar una cantidad de recursos y diferenciar servicios se le conoce como *Calidad de Servicio (QoS, Quality of Service)*. Para que *Internet* tenga esta capacidad se han desarrollado dos soluciones básicas con diferentes formas

de atacar el problema. La primera de ellas se conoce como *Arquitectura de Servicios Integrados (IntServ)* y la segunda se denomina *Arquitectura de Servicios Diferenciados (DiffServ)*.

2.3 Tecnologías para el soporte de QoS

En la actualidad, el soporte de *Calidad de Servicio* está basado principalmente en dos arquitecturas estándar: La arquitectura de *Servicios Integrados (IntServ)*[19] y la *Arquitectura de Servicios Diferenciados (DiffServ)*[20]. La *Arquitectura de Servicios Integrados* es utilizada principalmente en *Redes de Acceso* debido a que se adapta fácilmente a las necesidades de recursos de los usuarios pero a su vez tiene problemas de escalabilidad debido al agotamiento de los recursos de la red. Por otro lado, la *Arquitectura de Servicios Diferenciados* es muy escalable (soporta una gran cantidad de usuarios) pero a cambio, no puede adaptarse fácilmente a las necesidades de recursos de los usuarios. Por tanto, *DiffServ* es utilizada principalmente en *Redes de Transporte*. Una tercera parte dentro de este escenario son las redes *MPLS (Multi-Protocol Label Switching)* [21] que soportan los principios de Ingeniería de tráfico. Adicionalmente, *MPLS* puede complementarse con la *Arquitectura de Servicios Integrados* o con la *Arquitectura de Servicios Diferenciados* para soportar *QoS* de una mejor manera en una *Internet*. Estas tecnologías se explicarán a continuación.

2.3.1 Arquitectura de servicios Integrados

En esta aproximación [19] se hace reserva de recursos por flujos. Un flujo es una cadena de paquetes que fluyen por la red desde una aplicación en un ordenador origen hasta una aplicación en un ordenador destino. La reserva de recursos debe establecerse previamente en cada uno de los *Encaminadores* que hacen parte del camino entre los dos terminales. Para ello cuando una aplicación desea iniciar una comunicación debe seguir los siguientes pasos:

- a. La fuente inicia el establecimiento de una reserva describiendo primero a la red las características del flujo y los requisitos de los recursos.

- b. La red puede aceptar este nuevo flujo de aplicación sólo si hay suficientes recursos para comprometerse con los recursos solicitados.
- c. Una vez la reserva es establecida, la aplicación puede enviar sus paquetes a lo largo del camino reservado y la red cumplirá su compromiso.

IntServ trabaja bajo el supuesto de que la red tiene más recursos de los que se le solicitan. En la práctica esto no se cumple siempre pues en la medida que la red reserva recursos estos no estarán disponibles para otras comunicaciones, por lo que *IntServ* presenta problemas de agotamiento de recursos. El principal parámetro de Calidad del Servicio con el que se compromete *IntServ* es el *Retardo por Paquete*, específicamente el *Límite de Retardo del Peor Caso*.

El modelo de referencia de *IntServ* [22] (Figura 2.4) puede ser dividido en dos partes: (a) El plano de Control, que establece la reserva de recursos, y (b) El plano de datos, que re-envía los paquetes de datos basado en el estado de la reserva.

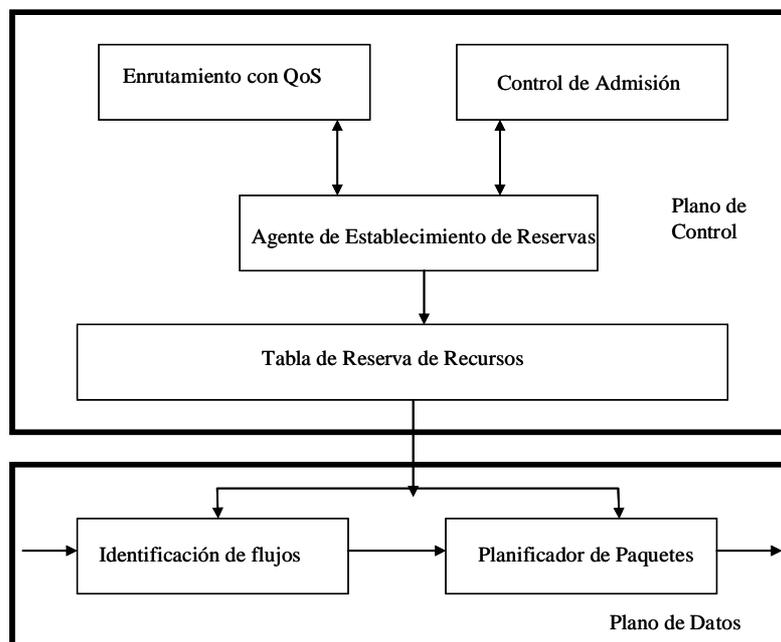


Figura 2.4 Modelo de referencia de *IntServ*

El Plano de Control (Control Plane): Para establecer la reserva de recursos, una aplicación primero caracteriza su flujo de tráfico y especifica los requisitos de *QoS*. A este proceso se le llama en *IntServ*: *Especificación del Flujo (Flow Specification)*. La

solicitud de establecimiento de reserva de recursos es entonces enviada a la red. Cuando un *Encaminador* recibe la solicitud, realiza dos tareas:

- a. Interactúa con el módulo de Encaminamiento para determinar el siguiente salto al que debe ser enviada la solicitud de reserva.
- b. Tiene que coordinar con el *Control de Admisión* para decidir si hay suficientes recursos para comprometerse con los recursos solicitados.

Para realizar el establecimiento de la reserva a lo largo del camino de los paquetes es necesario utilizar un protocolo de establecimiento de reserva de recursos. El protocolo que usa *IntServ* para este efecto es *RSVP (Resource Reservation Protocol)* [23]. Una vez completado el establecimiento de la reserva, la información del flujo reservado es instalada en la *Tabla de Reserva de Recursos*. Esta información es usada para configurar el módulo de *Identificación de Flujos (Flow Identification)* y el módulo de *Planificación de Paquetes (Packet Scheduler)* en el *Plano de Datos*.

El Plano de Datos (Data Plane): Cuando llegan los paquetes al *Encaminador*, el módulo de *Identificación de Flujos* (también llamado *Clasificador*) selecciona los paquetes que pertenecen a los flujos reservados y los coloca en las colas apropiadas. A su vez, el *Planificador de Paquetes* asigna los recursos a los flujos basado en la información de la *Tabla de Reservas de Recursos*. El procedimiento de *Clasificación* será descrito en detalle en el capítulo 3 debido a que en este procedimiento se ubica uno de los principales cambios en que se fundamenta la propuesta objeto de esta tesis. Por otro lado, el proceso de Planificación será explicado más adelante en este capítulo.

2.3.1.1 Modelos de Servicio en IntServ

Los *Modelos de Servicio* describen las interfaces entre la red y sus usuarios en una arquitectura de asignación de recursos; esto es, cuáles son los servicios que los usuarios pueden solicitar a la red y que tipo de compromiso puede ofrecer la red frente a los recursos reservados. Los *Servicios Integrados* poseen dos modelos de servicio: el *Servicio Garantizado* y el *Servicio de Carga Controlada*[22].

2.3.1.1.1 Especificación de flujos

Para hacer una reserva de recursos, una aplicación debe caracterizar el tráfico que va a ser inyectado a la red y especificar los requisitos del servicio para el flujo. En los *Servicios Integrados* estos detalles son descritos en la llamada *Especificación del Flujo*. La *Especificación de Flujo (Flowspec)* es en esencia un contrato de servicio que especifica el tráfico que la fuente enviará y los recursos y servicios que la red se promete a entregar. Si la fuente viola su descripción de tráfico (por ejemplo, enviando una tasa más alta que la acordada), la red obviamente no podrá mantener sus promesas. Típicamente el tráfico es vigilado antes de su entrada a la red para asegurarse que este está siguiendo lo acordado.

Las características de un flujo pueden ser especificadas de muchas formas; la forma exacta depende de qué mecanismo de *Control de Admisión* y que *Planificador de Paquetes* se use. Los parámetros más comunes para describir las características de los flujos son:

- *Tasa Pico o máxima*: Es la tasa máxima a la cual una fuente puede generar tráfico. Está dada en bits/s.
- *Tasa Promedio*: Es la tasa de transmisión media sobre un intervalo de tiempo y está dada en bits/s. La tasa promedio puede ser calculada de muchas formas, y los resultados pueden ser diferentes. Es importante conocer el método exacto y el intervalo de tiempo usado en el cálculo. Típicamente la *Tasa Promedio* es calculada con una ventana de tiempo que puede desplazarse de tal forma que el período de tiempo utilizado para el cálculo pueda empezar desde cualquier instante deseado.
- *Tamaño de Ráfaga (Burst size)*: Es la máxima cantidad de datos que pueden ser inyectados dentro de la red a la tasa máxima. El *Tamaño de Ráfaga* refleja las explosiones de la fuente de tráfico. Para evitar pérdidas de paquetes en ráfagas de tráfico cuyo *Tamaño de Ráfaga* sea menor que el *Burst Size*, un *Encaminador* debe tener asignado para la fuente de tráfico un *Buffer* más grande que el *Burst-Size*.

En los *Servicios Integrados*, el tráfico se describe en términos de los parámetros de un algoritmo conocido como *Leaky Bucket* [22]. Una interpretación de estas palabras en español podría ser *Balde Goteante*. Algunos autores se refieren al esquema de *Leaky Bucket* como un *Esquema de Cesta de Testigos* ó *Token Bucket* (ó también *Colector de Testigos*) [24]; sin embargo ambos algoritmos se diferencian en que el *Leaky Bucket* impone restricciones fuertes en la tasa de transmisión de datos, mientras que el *Token Bucket* permite cierto nivel de comportamiento a ráfagas y a la vez impone un límite en la tasa media de transmisión de datos. Un *Leaky Bucket* (ver Figura 2.5) es una clase de regulador de tráfico y tiene dos parámetros: r , la *Tasa de Llegada de Testigos* (*Tokens Arrival Rate*), y b , la *Profundidad del Colector* (ó simplemente *Bucket*). En un regulador *Leaky Bucket*, los *Testigos* caen a una tasa constante r dentro del sistema, y son consumidos por los paquetes que van llegando. Cuando un paquete llega, el regulador enviará el paquete solo si el *Colector* tiene suficientes *Testigos*. Cuando un paquete deja el *Leaky Bucket*, el regulador remueve una cantidad de *Testigos* equivalentes al tamaño del paquete de salida. Si el paquete llega y no hay suficientes *Testigos* disponibles, el paquete es almacenado en el *Buffer* de paquetes, a la espera de más *Testigos*.

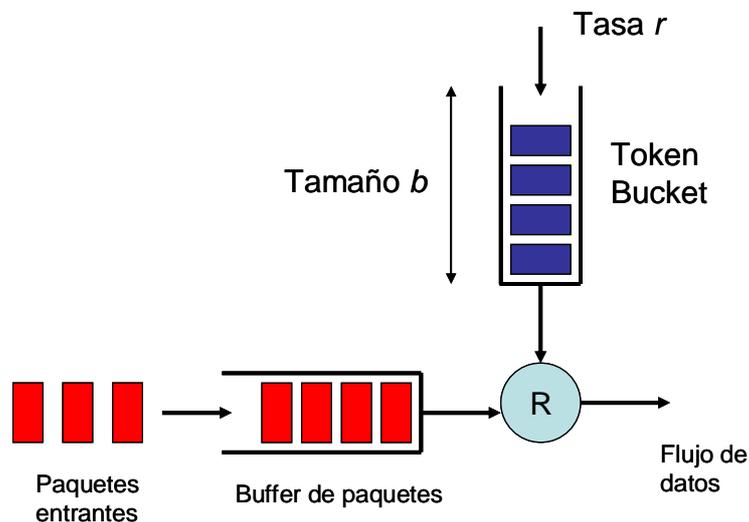


Figura 2.5 Operación del *Leaky Bucket*.

Cuando el nivel de *Testigos* acumulados llegue a la profundidad b , que es el límite de la máxima cantidad de *Testigos* que pueden ser acumulados, el regulador puede descartar

más *Testigos* hasta que el tamaño del *Colector* de *Testigos* sea menor que la profundidad b .

El regulador *Leaky Bucket* tiene un número de propiedades interesantes. Primero, el número total de bits que un *Leaky Bucket* permite que envíe una fuente está limitado por una función lineal. Supóngase que $A(t)$ es la cantidad de bits transmitidos durante cualquier intervalo de longitud t ; entonces se cumple que $A(t) \leq r \cdot t + b$. Segundo, la *Tasa de Llegada de Testigos* (r) corresponde a la *Tasa de Tráfico Promedio de Largo Plazo*. Tercero, la fuente puede generar variaciones de tráfico hacia la red, pero el tamaño máximo de estas no puede ser más largo que la profundidad del *Colector* (b). Cuando la fuente transmita una tasa menor que la *Tasa de Testigos* (r), los *Testigos* empezarán a acumularse dentro del *Colector*, que eventualmente se llenará. Si la fuente luego envía una ráfaga de paquetes, una gran cantidad de estos paquetes podrán pasar el regulador tan rápido como el enlace de salida lo permita.

2.3.1.1.2 Servicio garantizado

El *Servicio Garantizado* provee garantías de extremo a extremo de ancho de banda y límites estrictos de retardo en cola para sus flujos. El servicio está orientado a aplicaciones que requieran el más alto aseguramiento de ancho de banda y retardo; por ejemplo las aplicaciones de *Playback* intolerantes. El *Servicio Garantizado* también puede ser utilizado para aplicaciones que tienen requisitos de tiempo real fuertes.

Intuitivamente, el comportamiento de extremo a extremo de un trayecto que soporta el *Servicio Garantizado*, puede ser visto como un *Circuito Virtual* con un ancho de banda garantizado. El *Circuito Virtual* es más flexible que un circuito real; la partición del ancho de banda entre circuitos virtuales es lógica, y por tanto también los límites de las particiones permitiendo que se puedan expandir. Por ejemplo, el flujo de tráfico *Best-Effort* podría oportunamente usar el ancho de banda no consumido por los flujos reservados.

El *Servicio Garantizado* también provee límites estrictos de retardo. No controla el retardo mínimo o el retardo medio sino el *Retardo Máximo en Cola* (*Maximum Queuing*

Delay). La garantía del *Retardo Máximo en Cola* presupone que el tráfico entrante cumple con los parámetros especificados en el *Colector de Testigos*. Entonces las aplicaciones deben tomar en cuenta los retardos adicionales como el retardo de propagación de la trayectoria y el retardo de *Acondicionamiento de Tráfico* (retardo debido al acondicionamiento en el *Colector de Testigos*). En general, es difícil determinar los parámetros del *Colector de Testigos* para cualquier flujo de tráfico. A no ser que los parámetros del *Colector de Testigos* sean escogidos de una forma conservadora, algunos paquetes del flujo de una aplicación podrían violar los parámetros de este *Regulador*. Esto puede resultar en una formación de retardo extra en este proceso.

2.3.1.1.3 Servicio de carga controlada

Tanto el mantenimiento estricto del ancho de banda, como las garantías del retardo límite que son proporcionados por el *Servicio Garantizado* tienen un precio: los recursos tienen que ser reservados para el peor caso. Para fuentes de tráfico fluctuante, esto conllevará a una baja utilización de la red y a incrementar el costo por reserva de recursos. Además, es usualmente difícil conocer exactamente el ancho de banda y los requisitos de retardo para una aplicación.

Para algunas aplicaciones, un modelo de servicio con menos garantías estrictas y bajo costo podría servir mejor las necesidades. El grupo de trabajo de los *Servicios Integrados* consideró varias ofertas y estandarizó el *Servicio de Carga Controlada*. El *Servicio de Carga Controlada* no provee ninguna garantía cuantitativa sobre los límites de retardos ni del ancho de banda. En lugar de eso trata de emular una red de carga ligera para las aplicaciones que requieran el servicio. Este modelo de servicio permite multiplexación estadística y por tanto puede ser implementado de una forma más eficiente que garantice el servicio. Sus características se ajustan bien con aplicaciones adaptables que requieren algún grado de aseguramiento de desempeño mas no limites absolutos.

El *Servicio de Carga Controlada*, intenta crear un modelo de servicio intermedio entre el servicio de mejor esfuerzo (*Best-Effort*) y el servicio garantizado a través de un

Control de Admisión apropiado y mecanismos de aislamiento de tráfico. Por esta razón, también es referido como el servicio *Better-Than-Best-Effort* (mejor que el *Best-Effort*). El comportamiento visible a las aplicaciones puede ser mejor descrito como el de una red tipo *Best-Effort* ligeramente cargada. En otras palabras:

- Un porcentaje muy alto de los paquetes transmitidos serán exitosamente entregados por la red a los receptores.
- El tiempo de espera en cola experimentado por un porcentaje muy alto de paquetes entregados no excede en grandes proporciones al máximo retardo.

El *Servicio de Carga Controlada* no hace uso de valores específicos para el control de parámetros como el retardo y las pérdidas. La aceptación de una solicitud para el *Servicio de Carga Controlada* implica que la red tiene suficientes recursos para acomodar el tráfico sin causar congestión. En esencia, el *Servicio de Carga Controlada* permite un valor ocasional de retardos o pérdidas. Sin embargo, la probabilidad de tales eventos debe ser suficientemente baja para que el retardo promedio de encolamiento y la tasa de pérdidas promedio sobre un periodo razonable se sigan acercando a cero.

2.3.2 Arquitectura de Servicios Diferenciados

2.3.2.1 Aproximación Básica

DiffServ [20] fue desarrollada en respuesta a la necesidad de métodos toscos pero simples de proveer diferentes niveles de servicio para el tráfico de *Internet* y para soportar diferentes tipos de aplicaciones y negocios específicos. *DiffServ* emplea un conjunto de bloques a partir de los cuales se puede construir una variedad de comportamientos agregados. Para su funcionamiento *DiffServ* divide el tráfico en unas pocas clases y los recursos se asignan con base en las clases (no a los flujos individuales como *IntServ*), lo que hace que esta arquitectura no sufra el problema de agotamiento de recursos de la red.

La garantía de recursos a flujos individuales se provee a través de planificación y control de admisión, en vez de una reserva por flujos. Para el control de tráfico *DiffServ* tiene dos tipos de *Encaminadores*: los *Nodos Frontera* y los *Nodos Interiores* (ver Figura 2.6). Sólo los *Nodos Frontera* clasifican tráfico y marcan paquetes, mientras que los *Nodos Interiores* usan las clases codificadas en la cabecera del paquete (llamadas *Clases de Re-Transmisión* ó *Forwarding Equivalence Classes*) para determinar el tratamiento de los paquetes. El tratamiento que reciben los paquetes según su clase se denomina *Tratamiento de Re-Transmisión* (*PHB* ó *Per-Hop Behaviour*). En *DiffServ* se especifican estos *Tratamientos de Re-Transmisión* y no servicios de extremo a extremo. Por tanto, los servicios pueden ser construidos combinando *Clases de Re-Transmisión* y *Control de Admisión*. Finalmente, es importante aclarar que *DiffServ* crea diferentes niveles de servicio y aseguramiento de recursos, pero no da garantías de ancho de banda absolutas, ni límites de retardo para flujos individuales [22].

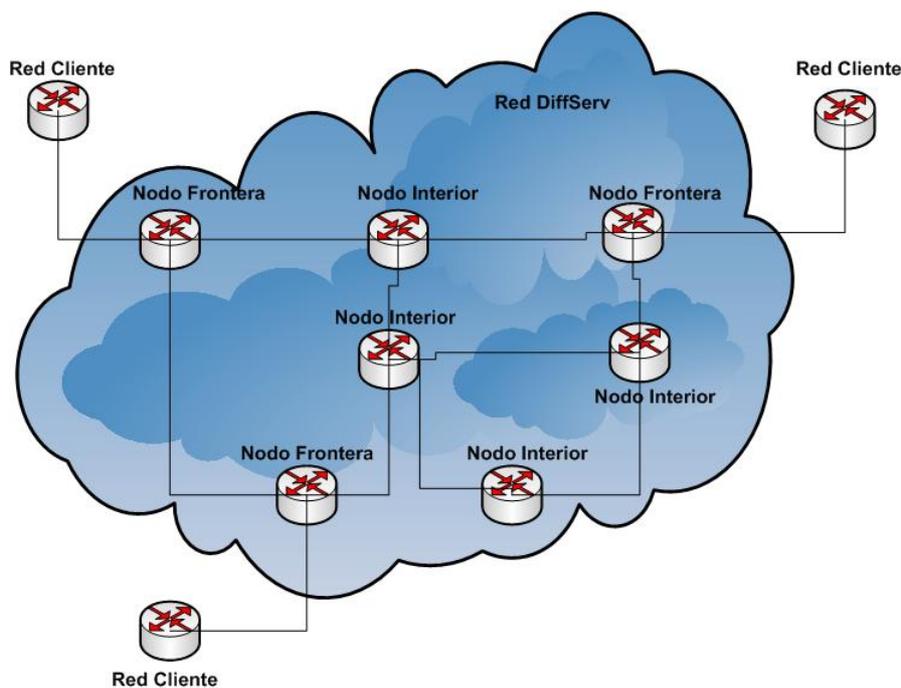


Figura 2.6 Arquitectura de una Red *DiffServ*

La aproximación de *Servicios Diferenciados* usa 6 bits en la cabecera del paquete *IP* para codificar el *Tratamiento de Re-Transmisión*, por lo que a continuación se describirá el campo de *Servicios Diferenciados* (conocido como *DS* ó *Differentiated Services*) descrito en los estándares de *Internet* [20].

2.3.2.2 Estructura del campo DS

La cabecera *IP* actual incluye un campo de 8 bits llamado *IP TOS*. Este está compuesto de un campo de *Precedencia* de 3 bits, un campo *Tipo de Servicio* de 3 bits (*TOS*, *Type of Service*) y dos bits que no usados y que deben ser cero (Figura 2.7). Los bits de *Precedencia* representan las prioridades para el tráfico, mientras que los bits *TOS* indican las preferencias para el *Tasa de Transferencia*, *Retardo* y *Pérdidas* (ver Tabla 2.1).

Precedencia	D	T	R	0	0
-------------	---	---	---	---	---

Figura 2.7 Campo *IP TOS*.

Tabla 2.1 Definición del campo *IP TOS*.

Bit	Descripción
0-2	Precedencia
3	0 = Retardo Normal 1 = Retardo Bajo
4	0 = Tasa de Transferencia Normal 1 = Tasa de Transferencia Alta
5	0 = Confiabilidad Normal 1 = Confiabilidad Alta
6-7	Reservado para uso futuro

El estándar de *Servicios Diferenciados* redefine el campo existente *IP TOS* para indicar los *Comportamientos de Re-Transmisión*. El nuevo campo, denominado *DS* (*Differentiated Services*), vuelve obsoletas las definiciones existentes del octeto *TOS* y también el octeto *Clase de Tráfico* de *IPv6*. Los primeros 6 bits del campo *DS* son usados como un *DSCP* (*Differentiated Services Code Point*), es decir, un valor que se utiliza para codificar el *PHB* con que debe tratarse un paquete en cada nodo *DiffServ* (Figura 2.8). Los restantes dos bits (campo *CU*) no están siendo utilizados actualmente. El *DSCP* debe tratarse como un índice, y la correspondencia entre los *DSCPs* y los *PHBs* debe ser configurable [22].

DSCP	CU
------	----

Figura 2.8 Campo *DS*.

2.3.2.3 *Gestión de Recursos Extremo a Extremo*

Los *Servicios Diferenciados* se enfocan en la gestión de recursos para un único *Dominio*, particularmente los *Dominios* de las *Redes de Transporte*. La experiencia muestra que esto permite promover el desarrollo de *Internet*. Sin embargo, la cuestión fundamental de la gestión de recursos extremo a extremo no está resuelta; los usuarios experimentan un buen servicio sólo cuando hay soporte de *QoS* en todo el camino entre el transmisor y el receptor [22].

Actualmente no existe acuerdo acerca de cómo obtener gestión de recursos de extremo a extremo. Se han propuesto varias aproximaciones, pero ninguna de ellas ha traspasado la fase de Propuesta. En esta sección se discuten brevemente algunas propuestas para ilustrar los requisitos en la gestión de recursos extremo a extremo.

Una propuesta es ejecutar *Servicios Integrados* sobre *Redes de Transporte* basadas en *Servicios Diferenciados* [25]. En este entorno, los *Servicios Integrados* operan en el borde de la *Internet*, tanto en *Redes Cliente* como en proveedores de servicios regionales, mientras que las *Redes de Transporte* están basadas en *Servicios Diferenciados*. Desde la perspectiva de los *Servicios Integrados*, las nubes de *Servicios Diferenciados* son enlaces virtuales conectando *Redes Clientes* de *Servicios Integrados*.

Otra aproximación para la gestión del ancho de banda entre *Dominios* es llamada *Bandwidth Broker (BB)*. La propuesta *BB* presenta una arquitectura para mantener la asignación de recursos entre diferentes proveedores *Troncales* de servicio. Un *BB* es un agente que hace el seguimiento del consumo de ancho de banda actual y aplica políticas

de utilización dentro de un *Dominio Administrativo*. Externamente, un *BB* establece unos acuerdos de servicio bilaterales con *BBs* vecinos para el tráfico que cruza las fronteras. En esta arquitectura, los *BBs* actúan como entidades de gestión centrales responsables de la gestión de los recursos de ancho de banda dentro de sus propios *Dominios* y también para la coordinación de la asignación de ancho de banda entre *Dominios*[26, 27].

2.3.3 *Multi Protocol Label Switching (MPLS)*

MPLS [21] ha surgido como una nueva tecnología importante para *Internet*. Esta tecnología representa la convergencia de dos aproximaciones fundamentalmente diferentes en redes de datos: el *Datagrama* y el *Circuito Virtual*.

MPLS utiliza una aproximación conocida como *Conmutación de Etiquetas (Label Switching)* en donde se usa una etiqueta corta de longitud fija insertada en la cabecera de los paquetes para re-transmitirlos. El propósito de la *Conmutación de Etiquetas* no es reemplazar el Encaminamiento *IP* sino expandir los servicios que proveen las redes *IP* mediante el ofrecimiento de la posibilidad de realizar *Ingeniería de Tráfico*, tener *QoS* garantizada y establecer redes privadas virtuales (*VPN- Virtual Private Network*). La *Conmutación de Etiquetas* puede ser usada para simplificar el proceso de re-transmisión de paquetes y para independizar los protocolos de re-transmisión, permitiendo implementar múltiples protocolos sobre el camino de los paquetes.

Entre las principales características del *MPLS* se destaca la división en los planos de control y de envío de datos. El *Plano de Control* se encarga básicamente de dos funciones: *Creación de Rutas*, que implica la construcción de las tablas de encaminamiento, y la *Señalización de las Rutas*. A su vez, el *Plano de Envío de Datos* es el encargado de la conmutación de paquetes a través del intercambio de etiquetas. Las etiquetas son insertadas al comienzo del paquete en la entrada de la red *MPLS*. En cada salto, el paquete es encaminado según el valor de la etiqueta y sale por el enlace correspondiente con otra etiqueta. Se obtiene una gran rapidez en la conmutación gracias a que las etiquetas son insertadas al principio del paquete y son de longitud fija, lo que hace que pueda hacerse una conmutación vía *Hardware* [21].

En *MPLS*, la transmisión de datos ocurre en los denominados *Label Switching Path (LSP)*. Los *LSP* son secuencias de etiquetas en cada nodo del camino desde la fuente hasta el destino. Hay dos formas de establecer los *LSPs*, la primera se realiza antes de la transmisión de datos (modalidad llamada también *Control-Driven*) y la segunda se hace una vez detectado un cierto flujo de datos (modalidad *Data-Driven*). Las etiquetas se distribuyen utilizando protocolos de señalización, los tres principales recomendados por la *IETF (Internet Engineering Task Force)* son: *Label Distribution Protocol (LDP)*, *ReSource reSeRVation Protocol (RSVP)* [23] y el *Constraint-Based Routing label Distribution Protocol (CR-LDP)* [28]. Las especificaciones actuales no presuponen ni imponen ningún protocolo específico para distribuir las etiquetas entre los *Label Switched Router (LSR)*, pero cabe destacar que las etiquetas son siempre asignadas en el sentido destino-fuente. Dependiendo de como se establezcan los *LSP* se pueden presentar diversas opciones. Si se utiliza la aproximación paso a paso (*Hop by Hop*) para el establecimiento de los *LSP*, la *IETF* ha recomendado el uso del protocolo *LDP*. En este caso también se pueden utilizar los protocolos *RSVP* y *CR-LDP* [29, 30]. Si la estrategia utilizada es la *Downstream Unsolicited*, en la cual el *Label Edge Router (LER)* de salida distribuye las etiquetas que deben ser utilizadas para alcanzar un determinado destino, la única opción disponible es usar *CR-LDP*.

Una red *MPLS* esta compuesta por *Encaminadores MPLS: Label Switched Router (LSR)* que representan el núcleo de la red (*Backbone*) y los *Label Edge Router (LER)*, que son los encargados de realizar la interfaz con otras redes, como se observa en la Figura 2.9. Los *LSR* son *Encaminadores* de gran velocidad en el núcleo de una la red *MPLS*. Sus Principales funciones son: participar en el establecimiento de los *LSPs* usando un protocolo de señalización apropiado y conmutar rápidamente el tráfico de datos entre los caminos establecidos. Los *LER* son *Encaminadores* en la frontera de la red a los que se pueden conectar diversas redes: *Frame Relay*, *ATM*, *Ethernet*, etc. Un *LER* envía el tráfico entrante a la red *MPLS* utilizando un protocolo de señalización de etiquetas y distribuye el tráfico saliente entre las distintas redes. Este tipo de *Encaminador* se divide a su vez en: nodos de entrada (*Ingress Node*) y nodos de salida (*Egress Node*).

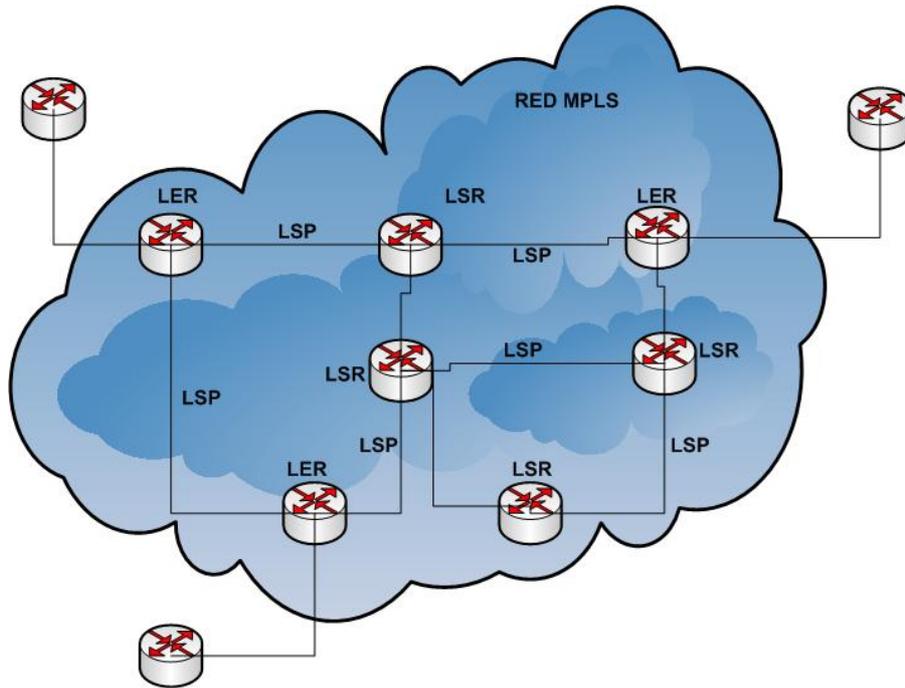


Figura 2.9 Arquitectura de una red *MPLS*.

Otra característica importante en *MPLS* es la posibilidad de hacer Encaminamiento explícito, el cual consiste en una ruta que ha sido configurada en el *LER* de entrada de acuerdo a los criterios de *QoS* y a la información de Encaminamiento de la red. *MPLS* clasifica en el *LER* de entrada el tráfico de modo que le asigna un *Forwarding Equivalence Class (FEC)*. Un *FEC* es un conjunto de paquetes que comparten unas mismas características para su transporte, así todos recibirán el mismo tratamiento en su camino hacia el destino. La asignación de un paquete a un determinado *FEC* se produce a través de la dirección destino o por medio de la clase de tráfico que es típicamente asociada con la dirección del puerto destino. Cada flujo de tráfico tendrá asociada una *FEC* que podrá asignarse a un *LSP* determinado.

Debido a que *MPLS* ofrece la posibilidad de realizar Encaminamiento explícito, señalizando cuales son los nodos que conforman un *LSP*, permite evitar situaciones de congestión. Las herramientas de *Agregación de Flujos* y *Encaminamiento Explícito* hacen de *MPLS* un protocolo ideal para realizar *Ingeniería de Tráfico*. Cuando se detecte un fallo en la red o la topología cambie, se debe de proporcionar un nuevo *LSP* para re-encaminar el tráfico [31]. En una ruta explícita estricta sólo se puede re-encaminar el tráfico en el *LER* de entrada, que es quien decide la ruta, por lo que debe

ser informado de los errores para proporcionar una ruta alternativa. En una ruta explícita tolerante cualquier *LSP* puede tomar un camino alternativo si es capaz de detectar el fallo de un *LSP* vecino, si la ruta ya está disponible o si un *LSP* de mayor prioridad requiere esos recursos reservados.

2.4 Procesos para el soporte de QoS

Para que una red *IP* soporte *QoS* mediante *Servicios Integrados* o mediante *Servicios Diferenciados*, sus *Encaminadores* deben realizar ciertos procesos que permitan dar tratamiento especial a los paquetes según el tipo de servicio al que se compromete la red. Algunos de estos procesos, de particular importancia para esta Tesis, se describen a continuación.

2.4.1 Establecimiento de Reservas: El protocolo RSVP

Para realizar las reservas, *IntServ* utiliza el protocolo *RSVP* (*Resource Reservation Protocol*). *RSVP* [23, 32]) fue diseñado para ser un protocolo agregado a la arquitectura de protocolos *TCP/IP* y su diseño fue influenciado por los requisitos de aplicaciones de video-conferencia *Multicast*. Los Principios de diseño de *RSVP* son:

- *Reservación simplex*: Las reservas se establecen en un solo sentido. Para comunicaciones en dos sentidos se debe establecer una reserva diferente para cada uno de ellos.
- *Es orientado al Receptor*: Ya que *RSVP* puede operar en comunicaciones *Multicast*, las cuales deben soportar grandes grupos con características de pertenencia dinámica y gran diversidad de tipos de receptores, se determinó que los receptores son responsables de decidir qué recursos serán reservados y deberán iniciar la reserva. La solicitud de reserva viaja desde el receptor hacia el transmisor y construye gradualmente el árbol de reserva
- *Independencia del Encaminamiento*: *RSVP* está diseñado para operar con protocolos de Encaminamiento *Unicast* y *Multicast* actuales y futuros, por lo

que la decisión de seleccionar un camino para un flujo es hecha por el protocolo de Encaminamiento; *RSVP* simplemente consulta la tabla de Encaminamiento y envía los mensajes *RSVP* de acuerdo con ella.

- *Independencia de las políticas:* El control de los parámetros que son transportados por *RSVP* es realizado por otros módulos de control de *IntServ*.
- *Adaptabilidad a cambios en la red:* *RSVP* debe adaptarse a los cambios en los miembros del árbol *Multicast* al igual que a cambios en la topología. Es por eso que utiliza un temporizador de caducidad de la reserva, que al vencerse borra la reserva existente. Es por ello que deben realizarse refrescos periódicos del estado de la reserva.
- *Varios estilos de reserva:* *RSVP* tiene varios tipos de reserva que son tratados de diferente forma en *IntServ*. Estos tipos de reserva permiten compartir una reserva para flujos de tráfico desde múltiples transmisores o seleccionar un transmisor particular en que está interesado un receptor

El procedimiento para el establecimiento de reservas con *RSVP* puede observarse en la Figura 2.10. Para establecer una reserva mediante *RSVP*, el *Host* de origen debe enviar un mensaje *PATH* que indica a los *Encaminadores* que conforman el camino de los paquetes cuáles recursos deben reservarse. En cada *Encaminador* del camino se analiza si se tienen suficientes recursos y, de ser suficientes, se re-envía el mensaje *PATH* al siguiente *Encaminador* en el camino de los paquetes. Una vez llega el mensaje *PATH* hasta el *Ordenador Destino*, este define exactamente qué recursos requiere y envía un mensaje de confirmación (*RESV*) hacia el *Ordenador Origen* por el mismo camino que siguieron los mensajes *PATH*. En cada *Encaminador* del camino, se reservan los recursos solicitados en el mensaje *RESV*. El procedimiento de establecimiento de las reservas finaliza cuando el mensaje *RESV* llega hasta el *Ordenador Origen* y a partir de ese momento, éste puede enviar el flujo de paquetes de información.

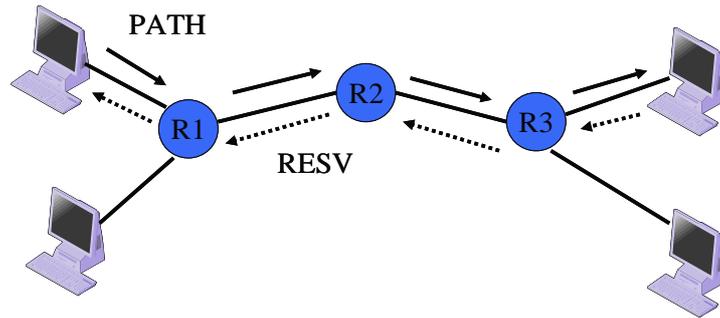


Figura 2.10 Establecimiento de reservas con el protocolo *RSVP*.

Los mensajes *RSVP* son enviados salto a salto entre *Encaminadores* (con capacidad *RSVP*) en forma de *Datagramas IP* pero usando el 46 como identificación del protocolo. Los mensajes *PATH*, *PATH Tear* y *ResvConf* no tienen como dirección destino el siguiente salto por lo que se envían con la opción *Routing Alert* en la cabecera del paquete *IP* ya que los *Encaminadores* dan un tratamiento especial a los paquetes que llevan esta opción[23].

El formato de los mensajes *RSVP* [23](ver Figura 2.11) consta de una cabecera común *RSVP* y un objeto a transportar. Dicho objeto consta a su vez de una cabecera común de objeto y del contenido del objeto mismo. Adicionalmente, en la Figura 2.12 se observa el formato de la cabecera común *RSVP* y de la cabecera del objeto y en la Tabla 2.2 se muestra un listado de los mensajes *RSVP* y su función.

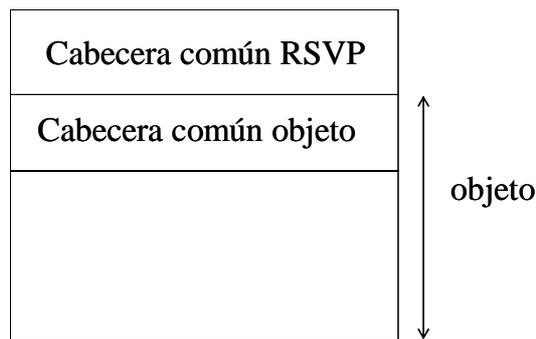


Figura 2.11 Formato de los mensajes *RSVP*.

Version	Banderas	Tipo de mensaje	Verificacion de suma RSVP
TTL de envío		Reservado	Longitud RSVP

Cabecera común

Longitud	Clase (Class)	Tipo (C-type)
Contenido del Objeto (longitud variable)		

Cabecera de objeto RSVP

Figura 2.12 Formato de la cabecera Común *RSVP* y de la cabecera de objeto *RSVP*.

Tabla 2.2 Tipos de mensajes *RSVP*

Tipo de mensaje	Descripción
PATH	Solicitud de reserva
RESV	Mensaje de reserva
PATHErr	Respuesta de error a PATH
RESVErr	Respuesta de error a RESV
PATH Tear	Tumbar el camino PATH
RESV Tear	Tumbar el camino RESV
RESVConf	Mensaje de Confirmación de Reserva

Uno de los objetos más importantes, por ser el que transporta la información de las características de la reserva es el objeto *SenderTspec*. El formato de dicho mensaje se observa en la Figura 2.13 y transporta entre otros, datos tales como la *Tasa de Testigos*, el tamaño del *Colector de Testigos*, la *Tasa Pico* de datos y el *Tamaño Máximo de Paquete*.

version	reservado	Longitud total
Cabec. reserva	reservado	Longitud de servicio
Tasa de Tokens (r)		
Tamaño del Bucket (b)		
Token Bucket Rate (r)		
Tasa pico de datos (p)		
Unidad de admisión mínima (m)		
Tamaño máximo de paquete (M)		

Figura 2.13 Formato del objeto *SenderTspec*.

2.4.2 Clasificación de flujos

Generalmente, la *Clasificación de Flujos* es el primer paso de procesamiento que sufre un paquete al llegar a un *Encaminador*. Una vez clasificado el paquete, un *Encaminador* puede proveer diferentes niveles de servicio a los paquetes de cada flujo. A continuación se describirá este proceso de gran importancia en la propuesta descrita en esta Tesis.

2.4.2.1 Definición y tipos de clasificación

La *Clasificación de Flujos* es el proceso de dividir un flujo de paquetes entrantes en un conjunto de flujos de paquetes relacionados. Una vez clasificados, un *Encaminador* puede proveer diferentes niveles de servicio a los paquetes de cada flujo. Generalmente, la *Clasificación de Flujos* es el primer paso de procesamiento que sufre un paquete al llegar a un *Encaminador*. La función de un *Clasificador* es generalmente es atravesar los planos de datos y control. El *Plano de Control* es responsable de crear y mantener una *Tabla de Flujos* que es usada por el *Plano de Datos* para clasificar paquetes. Cuando se establece un nuevo flujo, el *Plano de Control* inserta un nuevo *Registro de Flujo* en la *Tabla de Flujos*. Cuando llega un paquete, el *Plano de Datos* obtiene el *Registro del Flujo* usando la información contenida en los paquetes[33, 34].

Hay dos modelos para identificar un paquete que pertenece a un flujo: (1) *Tagging*, donde cada paquete lleva un campo explícito que identifica el flujo (es decir, un campo que contiene un índice que apunta a una *Tabla de Flujos*); y (2) *Matching*, donde se combinan varios campos contenidos en el paquete para identificar un flujo (por ejemplo, direcciones y números de puerto de origen y destino, y el campo de Identificación del protocolo). *Tagging* y *Matching* ofrecen un compromiso para las arquitecturas de red. El *Tagging* es de implementación simple en un *Encaminador*; sin embargo, la red necesita soportar un protocolo de señalización extremo a extremo (por ejemplo, como en *MPLS*) que permita a los *Encaminadores* acordar las *Etiquetas de Flujo*. El *Matching Multi-Campo*, en general, es más complejo; este podría involucrar búsquedas de coincidencia por rangos, prefijos o valores exactos en los valores de los campos. Un caso simple pero muy importante de *Matching Multi-Campo* es la clasificación basada en *Hashing*. En este tipo de solución, los *Encaminadores* derivan una *Clave Hash* con base en varios campos de la cabecera del paquete y usan esta clave como el *Identificador de Flujo* para encontrar el *Registro del Flujo*[34, 35].

2.4.2.2 *Tagging vs. Matching*

En el *Tagging*, un campo en la cabecera del paquete contiene la *Identificación del flujo* (*Flow ID*) para el paquete; un *Encaminador* puede utilizar este campo *Flow ID* para obtener el correspondiente *Registro del Flujo* de la *Tabla de Flujos*.

Para implementar el *Matching* se puede usar un esquema basado en *Hashing* con una *Tabla Hash* con su correspondiente *Tabla de Colisiones* para resolver tal situación. Por tanto, para identificar un flujo se utiliza la *Quíntupla* (*Dirección IP fuente, Dirección IP Destino, Puerto Fuente, Puerto Destino, Identificación del Protocolo*). Esto permite al clasificador hacer una identificación con una granularidad fina pero presentando problemas de escalabilidad. Hay dos razones para utilizar el *Hashing* como mecanismo de identificación de flujos: Primero, el algoritmo de *Hashing* básico es simple y puede implementarse eficientemente usando funciones de *Hashing* soportadas en *Hardware*; segundo, los esquemas de *Hashing* tienen un buen desempeño medio, lo que permite liberar recursos de *Hardware* para otras tareas. Dos esquemas comunes de *Hashing* son

el *XOR Folding* y el *CRC-32*. El primero es simple y consiste en encadenar todos los campos de la *Quíntupla* en una sola cadena y luego partirla en palabras de tamaño fijo (correspondiente al tamaño de la *clave hash*) con las que se realiza una operación *XOR* bit a bit entre ellas. En caso de que la longitud de la cadena completa no sea múltiplo del tamaño de la *Clave Hash*, la última palabra debe completar con ceros los bits que falten. Por otra parte, el esquema *CRC-32* es el conocido algoritmo para detección y corrección de errores ampliamente usado en sistemas de comunicación. Ambos esquemas presentan un desempeño similar [22].

Es importante aclarar que un *Encaminador* puede diferenciar entre paquetes que pertenecen a varios flujos sólo después de que el paso de clasificación *Demultiplexa* los paquetes entrantes. Por tanto, para *Encaminadores* que proveen *QoS*, el paso de clasificación debe operar a velocidades de línea. De otra manera, la tarea de clasificar un paquete de baja prioridad puede violar los requisitos de *QoS* de paquetes con alta prioridad. Debido a que los tiempos de acceso a memoria constituyen un costo predominante de la implementación de estos algoritmos, la necesidad de operar a velocidades de línea pone un límite al máximo número de accesos de memoria que pueden ser realizados al clasificar un paquete. Este requerimiento tiene una implicación para el esquema de clasificación basado en *Hashing*. En el peor caso, un gran número de flujos podría requerir leer el mismo registro de la *Tabla de Flujos*, requiriendo así, varios accesos a memoria. Por tanto, debido al límite máximo de accesos a memoria, podría llegar el caso de rechazarse un flujo durante la clasificación por haberse alcanzado este límite. Por otro lado, en el caso de *Tagging*, el caso medio y el peor caso son el mismo, así, sólo se rechazaría un flujo si la *Tabla de Flujos* está llena.

2.4.3 Planificación de Paquetes

El *Planificador de Paquetes* es el responsable de asegurar la asignación de recursos a flujos individuales y decide qué paquetes deberían tomar los recursos [22]. Un planificador puede verse como un despachador ya que mantiene información de cuántos paquetes ha enviado cada flujo y los compara con la cantidad de recursos que este ha

reservado. Los paquetes de un flujo son enviados sólo cuando no se han usado todos los recursos reservados por este.

2.4.3.1 Requisitos Básicos de un Planificador

Los requisitos básicos de un planificador pueden resumirse en tres aspectos: Aislamiento y distribución de los paquetes, Límites de retardo de los paquetes y Asignación del ancho de banda. Estos aspectos se detallan a continuación.

➤ *Aislamiento y Distribución*

El propósito de un planificador es permitir compartir un recurso común de forma controlada. En este ambiente hay dos aspectos importantes a tener en cuenta. El primero es el *Aislamiento de Flujos*, es decir, el grado de independencia entre los flujos que conlleva a que las alteraciones de un flujo no produzcan alteraciones a otros flujos. El caso extremo de la independencia sería el de la conmutación de circuitos. Sin embargo, en este caso los recursos reservados se sub-utilizan para el caso de la transmisión de datos. El segundo aspecto es la *Distribución de Recursos*. Esto se refiere al manejo eficiente de un recurso, lo que se logra con la multiplexación estadística usada en conmutación de paquetes. Por tanto, hay un compromiso entre estos dos aspectos. Una solución que logra un balance es la que se conoce como *Algoritmo de Distribución Justa (Fair-Queuing)*.

➤ *Límites de retardo*

Hay dos tipos de límites de retardo de los paquetes. Por una parte están los límites *Determinísticos* que requieren fuertes garantías de reservas. Un ejemplo de este tipo es el retardo del peor caso en *IntServ*. Por otro lado, están los límites *Estadísticos* que permiten *Multiplexación Estadística*. Ejemplos de este tipo de límites son los retardos por un tiempo máximo debido al crecimiento de las colas y el valor medio de retardo de los paquetes.

➤ *Asignación de ancho de banda*

Cuando no hay suficientes recursos para satisfacer las demandas de tráfico, debe asignarse de forma justa el ancho de banda para todos los flujos en competencia. Una política de distribución justa de recursos (*Criterio de Justicia*) que ha sido ampliamente considerada en la literatura es llamada *Max-Min Fair Sharing*.

2.4.3.2 Algoritmos de Planificación tipo Fair-Queuing

Los algoritmos de planificación [22] pueden ser clasificados en general en 3 categorías: *Algoritmos de Distribución Justa (Fair Queuing)*, *Algoritmos Basados en Plazos (Deadline Based)* y *Algoritmos Basados en Tasas (Rate Based)*. Por ser de nuestro especial interés debido a que es el tipo usado en el estándar de *Servicios Integrados*, nos enfocaremos en los algoritmos de tipo *Fair-Queuing*.

En la aproximación *Fair-Queuing*, el ancho de banda asignado a un flujo es representado por un número real, a menudo denominado *Peso*. Estos algoritmos tratan de asignar un *Ancho de Banda (BW)* proporcional a los flujos activos o represados con base en sus pesos. Si un flujo no consume todo el *BW* asignado, el *BW* restante es asignado a flujos represados en proporción a sus pesos. De esta forma, se le garantiza a un flujo que puede usar su *BW* asignado y podría usar más. Además, se puede proveer un límite de retardo extremo a extremo (*End-to-End ó E2E*). Por estas razones, este tipo de algoritmos se usan en *IntServ*.

El más conocido de los algoritmos de planificación de tipo *Fair-Queuing* es el *WFQ (Weighted Fair Queuing)* [36]. *WFQ* fue propuesto hace más de 10 años y soporta la asignación de *BW* y límites de retardo. Este algoritmo tiene variantes con diferentes compromisos entre complejidad y precisión y ha sido implementado ampliamente en *Encaminadores* para soportar *QoS*.

WFQ es descrito a menudo como un modelo de fluidos. En el modelo de fluidos se asume que el tráfico es infinitamente divisible y que un nodo puede servir múltiples

flujos simultáneamente. Sin embargo, en una red real los paquetes son procesados uno a la vez y por tanto, el tamaño del paquete afecta el sistema de colas. Por ser un algoritmo construido para funcionar en una red real, *WFQ* trata de adaptar el modelo de fluidos a un modelo Paquetizado.

➤ *GPS (Generalized Processor Sharing)*

Debido a que el modelo de fluidos presenta ciertas ventajas tales como que permite visualizar la operación de las disciplinas de colas más fácilmente y además es sencillo derivar resultados para un sistema paquetizado de un modelo de fluidos, primero se describirá el modelo de fluidos ideal, conocido como *GPS (Generalized Processor Sharing)*.

GPS [36] es un algoritmo tipo *Fair-Queuing* ideal que provee una distribución de recursos justa con el criterio *Max-Min Fair Share*. En esta aproximación, cuando una conexión (ó flujo) tiene datos en la cola, se dice que está en un estado *Backlogged* (espera). *GPS* trabaja bajo los siguientes supuestos:

Sean N el *Número de Flujos* siendo servidos por un servidor, R la *Tasa de Servicio* del servidor, ϕ_i el *Peso* que se le asigna a un flujo i -ésimo y $S(i, \tau, t)$ la *Cantidad de Datos Servidos* para el flujo i durante el intervalo (τ, t) . En el intervalo (τ, t) el flujo i recibe una parte justa mínima R_i proporcional a su *Peso* así,

Ecuación 2.1:

$$R_i = \frac{\phi_i}{\sum_{j=1}^V \phi_j} R$$

donde V es el conjunto de flujos en espera durante el intervalo.

Con *GPS*, los flujos que no están en espera (envían menos de su parte asignada) reciben servicios sin ninguna cola. Además, el *BW* restante es repartido a los flujos en espera en proporción a sus pesos. De esta manera se garantiza la mínima parte a cada flujo y el ancho de banda sobrante es repartido proporcionalmente a los flujos en espera, por lo que *GPS* obtiene una repartición justa del *BW*.

Por otro lado, *GPS* puede garantizar un límite de retardo de (b/r) , si la fuente de tráfico es restringida con un *Colector de Testigos* con *Tamaño de Ráfaga* b y *Tasa de Testigos* r de forma que,

Ecuación 2.2:

$$r \geq \frac{\phi_i}{\sum_{j=1}^V \phi_j} R$$

➤ *WFQ (Weighted Fair Queuing)*

WFQ (Weighted Fair Queuing) [36] es la aproximación paquetizada de la planificación *GPS* y también se conoce como *PGPS (Packet-by-packet Generalized Processor Sharing)*. *WFQ* trata de emular un sistema *GPS* calculando el *Tiempo de Salida* de un paquete (*Finish Time*) en un sistema *GPS*. Esta marca de tiempo se usa para planificar los paquetes. El *Tiempo de Salida* no es el tiempo de salida real de un paquete, sino que es un número que representa el orden de los paquetes en el sistema. Si un paquete *A* termina la transmisión antes que un paquete *B* en un sistema *GPS*, el paquete *A* será seleccionado para transmisión antes que el paquete *B* en el sistema *WFQ*. El cálculo de las marcas de tiempo en *WFQ* se realiza de diferentes maneras según sea el caso.

Caso 1: Suponiendo que todos los flujos están en espera todo el tiempo (siempre enviando a tasas más altas que su *BW* reservado), la marca de tiempo se calculará como,

Ecuación 2.3:

$$F_i^k = F_i^{k-1} + \frac{L_i^k}{\phi_i}$$

donde F_i^k es la marca de tiempo del paquete k del flujo i y se calcula como el *Tiempo de Salida* del paquete previo del mismo flujo más el tiempo necesario para transmitir el paquete actual. L_i^k es la longitud del paquete k del flujo i .

Caso 2: Los flujos podrían cambiar de espera a no-espera y viceversa, por lo que podría haber tiempo no usado entre dos paquetes. En este caso, el cálculo de la marca de tiempo será,

Ecuación 2.4:

$$F_i^k = \max[F_i^{k-1}, V(t)] + \frac{L_i^k}{\phi_i}$$

donde $V(t)$ es una función lineal del tiempo real t en cada uno de los intervalos de tiempo.

2.5 Movilidad en redes IP

Actualmente, las direcciones en *Internet* son utilizadas entre otros aspectos para la identificación del sistema final, además *TCP* usa las direcciones *IP* para seguir la pista del estado de la sesión interna entre los dos extremos. Las direcciones también se usan para encontrar la ruta entre los dos extremos (la ruta no es la misma en las dos direcciones ya que depende de la dirección del destino y de la congestión de la red). En este punto aparece un gran dilema: Un ordenador necesita una dirección *IP* estable para ser identificable en forma estable por otros ordenadores de *Internet*, sin embargo, si la dirección *IP* es estable, el Encaminamiento hacia el ordenador móvil es estable y los *Datagramas* siempre irán al mismo sitio. Por tanto no hay movilidad. Para dar solución a este problema han surgido varias propuestas que buscan solucionar el la movilidad de diferentes maneras y en diferentes situaciones. Así, existen propuestas que solucionan la movilidad en entornos denominados de *macro-movilidad*, es decir, cuando se cambia de una *Red de Acceso* a otra (la más aceptada hasta el momento se denomina *IP Móvil* ó *Mobile IP* [37, 38]), mientras que hay propuestas que operan en entornos de *micro-movilidad*, es decir, cuando se hacen cambios de *Encaminador de Acceso* dentro de una misma *Red de Acceso* (algunas de las propuestas más conocidas son: *cellular IP* [4] y *HAWAII* [5], *IP Móvil Jerárquico* ó *HMIP* [3] y *NETLMM* [39]). En general, los protocolos de *micro-movilidad* buscan disminuir la cantidad de señalización requerida por los protocolos de *macro-movilidad* cuando los *Encaminadores de Acceso* se encuentran muy cerca entre sí.

2.5.1 Solución de Macro-Movilidad: El protocolo IP Móvil

En primer lugar se estudiará la versión estandarizada para *IPv4* (llamada *MIPv4*, *Mobile IPv4*) [1] y luego estudiaremos la versión para *IPv6* ó *Mobile IPv6 (MIPv6)* [2].

2.5.1.1 El protocolo Mobile IPv4

Mobile IPv4 [38] considera dos tipos de redes conectadas a *Internet*:

- *Home Network: (Red Doméstica)* Esta es una red en la que se encuentra matriculado un ordenador móvil.

- *Visited Network: (Red Visitada)* Es una red a la que se conecta un ordenador móvil cuando no se encuentra en su *Red Doméstica* y le permite al ordenador móvil conectarse a *Internet* como un huésped.

La solución suministrada con *IP Móvil*, en sus dos versiones, consiste en que los ordenadores móviles siempre se identifican por una dirección fija denominada *Home Address (Dirección Doméstica)* sin tener en cuenta su punto actual de acceso a *Internet*. Mientras esté fuera de su *Red Doméstica*, cada ordenador móvil tiene una dirección adicional que identifica su localización actual: la *CoA (Care of Address) ó Dirección a Cargo* (también se conoce como *Dirección Temporal*). Así, los mensajes que llegan a la *Dirección Doméstica* original son re-enviados mediante un túnel a la *CoA*.

Ahora se describirán las entidades que intervienen en *Mobile IPv4* (ver Figura 2.14):

- *Mobile Node (MN)*: Es un ordenador móvil que cambia su punto de conexión de una red o subred a otra. Mantiene una dirección *IP* constante. En este documento también se le llama *Nodo Móvil*.

- *Correspondent Node (CN)*: Es el ordenador que se comunica con el *MN* a través de *Internet* (es el otro extremo de la comunicación). Puede ser fijo o móvil. En este documento también se le llama *Nodo Correspondiente*.
- *Home Agent (HA)*: También se conoce como *Agente Propio*. Es un *Encaminador* ubicado en la *Red Doméstica*. Todo *Datagrama* que vaya hacia el *Nodo Móvil* debe pasar por el *Agente Propio* y este lo reenvía al *MN*.
- *Foreign Agent (FA)*: Conocido también como *Agente Visitado*. Es un *Encaminador* ubicado en la *Visited Network*. Este colabora con el *Home Agent* para entregar los *Datagramas* a un nodo móvil que se encuentra conectado a la *Red Visitada*.

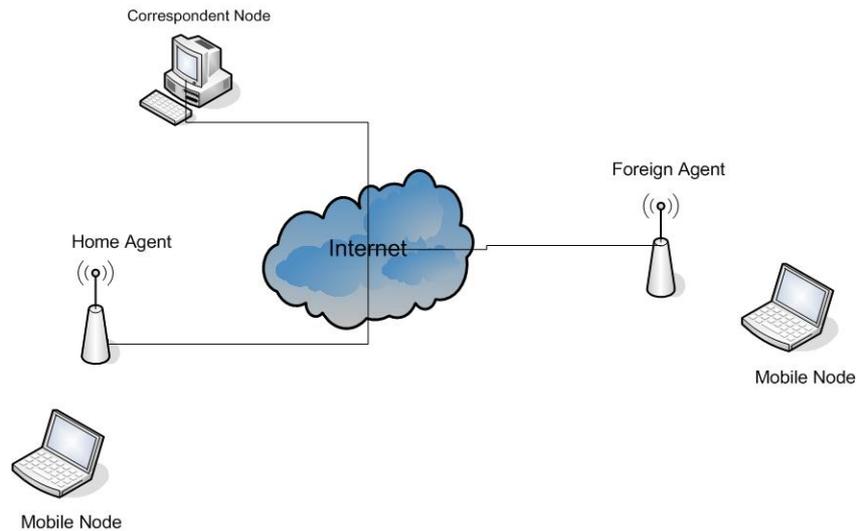


Figura 2.14 Entidades de *Mobile IPv4*.

De otra parte, *Mobile IPv4* tiene 3 funciones básicas [38] que son:

- *Descubrimiento de Agente*: Los *Agentes Móviles (HA y FA)* anuncian acerca de su disponibilidad sobre cada enlace en que se les provee servicio.
- *Registro*: Cuando un *MN* sale de su *Red Doméstica*, debe registrar su *Dirección Temporal (CoA)* con su *Agente Propio (HA)*.

- *Entunelamiento*: El HA envía los *Datagramas* a la CoA a través de un túnel (Usa *Encapsulado IP dentro de IP*[40] o también *Encapsulado Mínimo* [41]).

Estas funciones son combinadas para obtener la operación normal del protocolo *Mobile IPv4*. Durante la operación de *Mobile IPv4*, se tienen dos actividades importantes: el *Procedimiento de Registro* de un MN y el *Procedimiento de Envío de Datagramas* desde el CN. Un tercer aspecto importante es el *Procedimiento de Traspaso* (o *Handoff*) para cambiar de una subred a otra.

Para realizar el *Procedimiento de Registro* de un MN, tanto el HA como el FA envían *Mensajes de Anuncio* en su zona de cobertura para indicar su presencia a los MNs. Estos *Mensajes de Anuncio* llevan la dirección del *Agente Móvil* (HA o FA). Cuando el MN detecta este mensaje, puede determinar si se encuentra en su *Red Doméstica* o en una *Red Visitada*. En este último caso, el MN inicia un *Procedimiento de Registro* para obtener una CoA del *Agente Visitado* y luego registrar esta dirección ante el HA.

Para el *Procedimiento de Envío de Datagramas* desde el CN al MN, los *Datagramas* son enviados hacia el HA y luego este los envía por un túnel al *Agente Visitado*, quien a su vez los entrega al MN. Esta aproximación tiene una deficiencia, pues para ciertas situaciones el camino en el sentido CN a MN puede ser mucho más largo que el camino en sentido contrario. A este problema se le denomina el *Problema del Triángulo de Encaminamiento*. Para solucionar esto, se propone que el MN esté enviando unos mensajes de *Actualización de Vinculación* (BU, *Binding Update*). Estos informan la nueva CoA y son enviados hacia el *Agente Doméstico* y este a su vez le informa al CN de la nueva CoA del MN. Esto se hace para que el CN envíe los paquetes directamente a la CoA del MN. Este procedimiento es una situación extraordinaria en el protocolo *Mobile IPv4* y se denomina *Optimización de la ruta*.

Ahora se explicará el *Procedimiento de Traspaso*, conocido también como *Smooth Handoff*, pues cuando un MN se mueve de un punto de conexión de una subred a otro punto de conexión en otra subred dentro de *Internet*, es ideal que las transiciones sean lo más suaves posibles. La continuidad en la comunicación al cambiar de punto de conexión no es fácil pues los *Datagramas* pueden perderse en algún momento dentro

del tiempo en que se realiza el *Traspaso*. Esto se debe a que no hay forma de que todos los *Nodos Correspondientes* puedan recibir instantáneamente los *Mensajes de Actualización de Vinculación (Bindings)* que reflejen el movimiento del *MN*. Para solucionar este problema, los *FAs* almacenan la información de movilidad de sus antiguos *MNs* guardando la *CoA* actual de cada uno. Con esta información, un *FA* puede re-encapsular un *Datagrama*, que ha llegado a él por error durante el *Traspaso*, con la *CoA* correcta y enviarlo al *MN*. Los pasos que sigue el procedimiento de *Traspaso* son:

- Supongamos que el *MN* se encontraba anteriormente en una *Red Visitada* y ahora cambia a otra nueva *Red Visitada*.
- El *MN* realiza su *Procedimiento de Registro* y solicita al nuevo *FA* que envíe un mensaje de actualización de la nueva *CoA* al antiguo *FA*.
- El *MN* realiza la operación de registro ante el *HA*.
- Los mensajes que lleguen al antiguo *FA* hacia el *MN*, serán re-enviados a la nueva *CoA*.

Podría suceder que antes de que sea actualizada la nueva *CoA* en el antiguo *Agente Visitado*, llegase un *Datagrama* hacia el *MN*. En este caso habría un problema ya que el *FA* antiguo devolvería el *Datagrama* al *Agente Propio* y este a su vez lo devolvería al *FA* antiguo, creando de esta manera un lazo sin fin. Es por esto que en este caso el *FA* antiguo devuelve el *Datagrama* al *HA* pero con la dirección de *Origen* igual a la *CoA*. Así, cuando el *HA* detecta esto, sabe que debe esperar a que haya un nuevo *Registro* para reenviar el mensaje.

2.5.1.2 El protocolo Mobile IPv6

Mobile IPv6 [2] es muy similar a *Mobile IPv4*, la diferencia más grande radica en que, normalmente se realiza *Optimización de la ruta* en lugar del *Triángulo de Encaminamiento* para el Encaminamiento de los paquetes.

2.5.1.2.1 Mensajes de Mobile IPv6

Mobile IPv6 requiere el intercambio de información adicional. Los nuevos mensajes usados en *Mobile IPv6* son definidos como *Opciones de Destino* de *IPv6*. Por consiguiente, la información llevada sólo será examinada por el nodo de destino del paquete. Las nuevas *Opciones de Destino* definidas en *Mobile IPv6* son:

- *Actualización de Vinculación (BU, Binding Update)*: Un *MN* informa a su *HA* o a cualquier *CN* sobre su *CoA* actual usando la opción de *Actualización de Vinculación*.
- *Confirmación de Vinculación (BAck)*: Se usa para reconocer la recepción de una *Actualización de Vinculación* (si el reconocimiento fuera solicitado).
- *Solicitud de Vinculación (BR, Binding Request)*: Cualquier nodo, para solicitar a un *MN* enviar un *BU* con la actual *CoA*, usa la opción de *Solicitud de Vinculación*.
- *Dirección Doméstica (Home Address)*: La opción de la *Dirección Doméstica* se usa en un paquete enviado por un *MN* para informar al receptor de este paquete sobre la *Dirección Doméstica* del *Nodo Móvil*.

2.5.1.2.2 Estructuras de datos usadas en MIPv6

La especificación de *Mobile IPv6* describe las siguientes tres estructuras de datos:

Caché de Vinculaciones (Bindings Cache): Cada nodo tiene un *Caché de Vinculaciones* que se usa para almacenar las *Vinculaciones (Bindings)* de otros nodos.

Si un nodo recibe un *BU*, agregará esta *Vinculación* a su *Caché de Vinculaciones*. Así mismo, cuando quiera enviar un paquete, se busca una entrada en el *Caché de Vinculaciones*. En caso de que haya una entrada, el paquete se envía a la *CoA* del *Nodo Móvil* respectivo usando una *Cabecera de Encaminamiento de IPv6*.

Lista de BU's: Cada *Nodo Móvil* tiene una *Lista de BU's* que se usa para guardar la información sobre cada *Actualización de Vinculación* enviada por él y para las cuales el tiempo de vida no ha expirado aún. Esta lista contiene todos los *BUs* enviados a su *HA* y a cualquier *Nodo Correspondiente* (móvil o estacionario), es decir los nodos que se están comunicando o se han comunicado hace corto tiempo con el *MN*.

Lista de HA's: Para cada *Enlace Doméstico (Home Link)*, es decir, una *Red Doméstica*, un nodo que sirve como *HA* genera una lista que contiene la información sobre todos los otros *HA's* en esta red. La información en esta lista es aprendida de los *Mensajes Multicast de Anuncio* de los *Encaminadores*, los cuales son enviados desde todos los *HA*. La información acerca de los otros *Agentes Domésticos* es usada por el *DHADM (Dynamic Home Agent Discovery Mechanism)*.

2.5.1.2.3 Operación de Mobile IPv6

A continuación se explicarán los mecanismos de *Mobile IPv6*, para lo que se usará la Figura 2.15. La figura muestra tres enlaces y tres sistemas. En el *Enlace A (Red Doméstica)* reside un *Encaminador* que ofrece el servicio de *HA*. Este enlace es llamado también el *Enlace Doméstico* del *Nodo Móvil*. Este *Nodo Móvil (MN)* simplemente se mueve del *Enlace A* para unirse al *Enlace B (Red Visitada)*. Adicionalmente hay un *Nodo Correspondiente* sobre el *Enlace C* que puede ser móvil o estacionario. A continuación se detallarán los procedimientos de *Registro* y de *Envío de Datagramas* mediante el *Triángulo de Encaminamiento* y mediante la *Optimización de la Ruta*.

➤ *Registro de un HA:*

En cuanto un *Nodo Móvil* detecta que se ha movido de un *Enlace* a otro, y ha descubierto un nuevo *Encaminador*, el *MN* realiza una auto-configuración de la

dirección y usa la nueva dirección asignada como su *CoA*. El prefijo de esta *CoA* es el prefijo de la *Red Visitada* en que se encuentra ahora el *Nodo Móvil*; todos los paquetes serán dirigidos ahora a esta *CoA* y localizarán el *MN* en el *Enlace* actual.

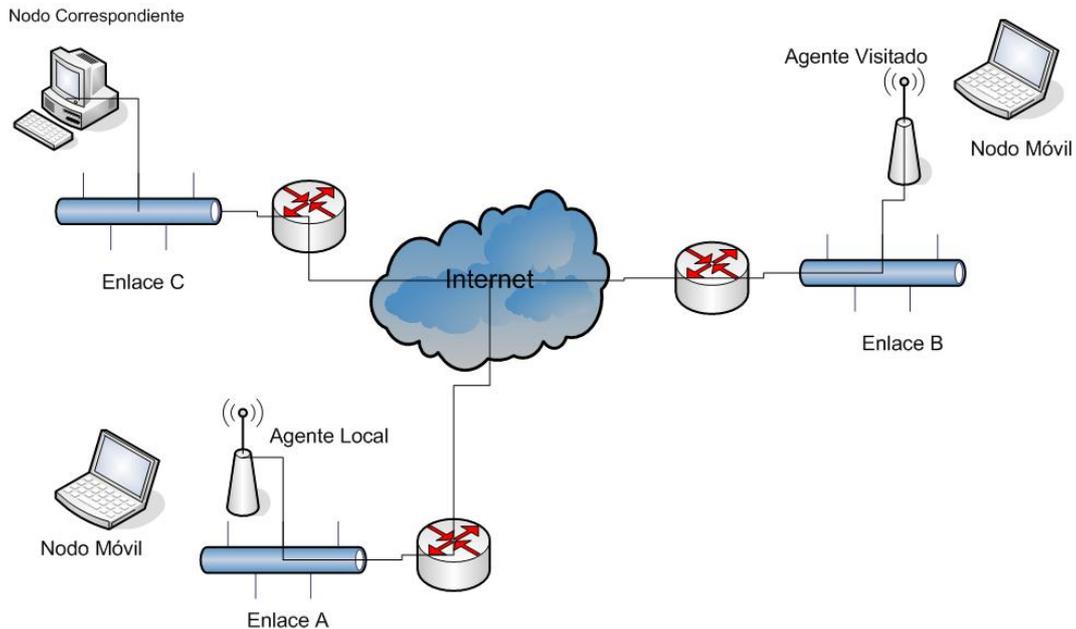


Figura 2.15 Escenario MIPv6

El *MN* registra su *CoA* con su *HA* en el *Enlace Doméstico*. Por consiguiente el *MN* envía un paquete, que contiene una opción *BU*, a su *HA*. El *HA* registra este enlace y retorna un paquete con una opción de *Confirmación de Vinculación (BACK)* al *Nodo Móvil*.

➤ *Triángulo de Encaminamiento:*

Una vez el *HA* haya introducido la *CoA* de un *MN* en su *Caché de Vinculaciones*, interceptará cualquier paquete enviado a la *Dirección Doméstica* del *Nodo Móvil*. Para ello, usa el procedimiento denominado *Proxy Neighbor Discovery (Descubrimiento de Vecinos)*. Cada paquete interceptado es encapsulado a la *CoA* registrada del *MN* usando *Encapsulado de IPv6* [42].

En sentido contrario, cuando el *MN* envía los paquetes a cualquier otro nodo, envía los paquetes directamente al destino. El *MN* pone en la *Dirección Origen* de este paquete la *CoA* e incluye una *Opción de Destino* de tipo *Dirección Doméstica*. Debido a que la

Dirección Doméstica es estática (en contraste con la *CoA*), es posible que cada *Nodo Correspondiente* use transparentemente la *CoA*. Las capas superiores (incluso las aplicaciones) no consideran la *CoA*, ellas sólo utilizan la *Dirección Doméstica*.

Si un *MN* se comunica con un *Nodo Correspondiente* que está fuera de su *Red Doméstica*, los paquetes son enviados desde el *CN* al *HA*, luego desde el *HA* al *MN* y en sentido contrario directamente desde *MN* al *CN*. Este es el ya mencionado *Triángulo de Encaminamiento*.

➤ *Optimización de la Ruta:*

Para evitar el problema del *Triángulo de Encaminamiento*, un *MN* puede enviar *BU's* a cualquier *CN* (Móvil o estacionario). Esto permite a los *CNs* almacenar la *CoA* actual y enviar los paquetes directamente a un *MN*.

Cualquier *Host IPv6* que envía un paquete verifica primero su *Caché de Vinculaciones* para esta *Dirección de Destino*. Si hay una entrada, enviará el paquete al *MN* usando una *Cabecera de Encaminamiento* (en lugar de usar *Encapsulado de IPv6*). La ruta especificada por esta *Cabecera de Encaminamiento* tiene dos saltos. El primer salto es la *CoA* y el segundo salto es la *Dirección Doméstica* del *MN*. Esto produce que el paquete sea enviado directamente a la *CoA* del *MN*. El *MN* recibe este paquete y lo envía al próximo salto especificado en la *Cabecera de Encaminamiento*. El próximo y último salto es la *Dirección Doméstica* del *MN*, por consiguiente este paquete se absorbe en el *MN*.

Si el *Caché de Vinculaciones* no tiene ninguna entrada, este paquete será enviado normalmente. Entonces este paquete se envía a la red especificada y es recibido por el *Nodo de destino*. En caso de que el destino sea un *MN* que está en una *Red Visitada*, este paquete será interceptado por un *HA* y será encapsulado al *MN*.

2.5.2 Soluciones de Micro-Movilidad

A continuación se estudiarán dos propuestas para solucionar los problemas de micro-movilidad que han sido acogidas por la IETF (Internet Engineering Task Force), el protocolo HMIP (Hierarchical Mobile IP) y la propuesta NETLMM (Network-Based Localized Mobility Management).

2.5.2.1 El protocolo Hierarchical Mobile IP

El protocolo *HMIPv6* [3] propone el uso de *Encaminadores* especiales denominados *MAP* (*Mobility Anchor Point*) ó *Puntos de Anclaje de la Movilidad*, los cuales tienen bajo su influencia un dominio compuesto por *Encaminadores de Acceso* (*AR*, *Access Router*) y *Nodos Móviles* (*MN*, *Mobile Node*) (ver Figura 2.16).

El *MAP* gestiona la movilidad del *MN* dentro de su *Dominio*, por lo que mientras el *MN* esté dentro de un mismo *Dominio MAP*, no se requiere comunicar al *HA* los movimientos de éste, ya que la señalización de la movilidad (*Envío de BUs*) sólo se requiere entre el *MN* y el *MAP* con el objetivo de informar al *MAP* del cambio de un *AR* a otro dentro del mismo dominio, por lo que el *MAP* sabrá a qué *AR* debe enviar los paquetes que van hacia el *MN*. Los paquetes viajan por un túnel desde el *MAP* hasta el *MN*.

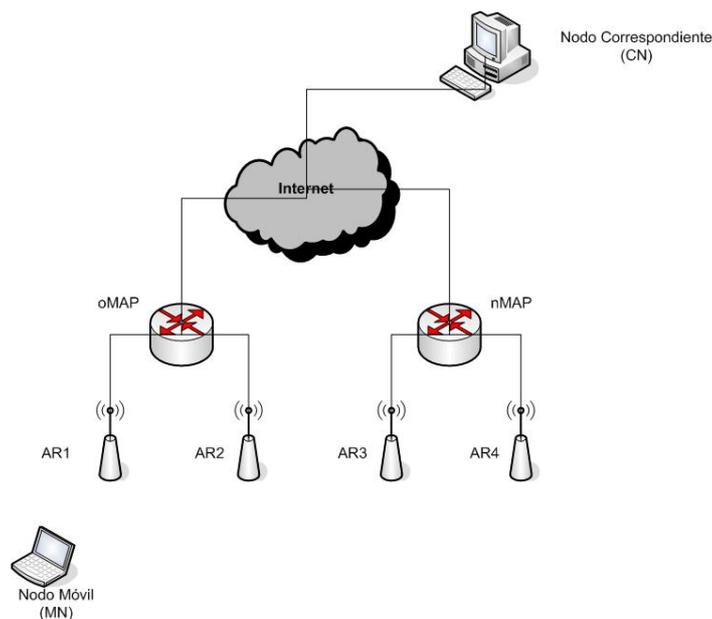


Figura 2.16 Arquitectura usada en *HMIPv6*.

Si un *MN* sale de un dominio *MAP*, debe hacer un proceso de *Traspaso* para cambiar de *MAP* y luego actualizar este movimiento con su *HA* mediante mensajes *BU* (*Binding Update*). Una vez actualizado el *HA*, este envía los paquetes del *MN* hacia el nuevo *MAP*, el cual se encarga de hacerlos llegar al *MN* mediante un túnel.

Para la gestión de la movilidad se emplean dos direcciones diferentes para un mismo *MN*. Una dirección denominada *lCoA* (*Local Care of Address*) se usa para la gestión local de la movilidad (dentro del *Dominio MAP*), mientras que otra dirección, denominada *rCoA* (*Regional Care of Address*) se usa para la gestión de la movilidad en *Internet*. Esta es la dirección que utiliza el *Nodo Correspondiente* para identificar el *MN* y está basada en el prefijo de subred del *HA*.

Para determinar si hay necesidad de un *Traspaso*, el *MN* está analizando continuamente los mensajes *RAdv* (*Router Advertisements*) enviados por los *ARs*. Estos mensajes llevan la identificación del *MAP* al cual está asociado el *AR* junto con la identificación del *AR*.

2.5.2.2 NETLMM

La propuesta NETLMM (Network-Based Localized Mobility Management) [39] busca manejar el problema de movilidad dentro del dominio de una red de acceso, mientras que deja el problema de manejo de movilidad entre redes de acceso a los protocolos de movilidad tales como MIP. La intención básica es descartar la necesidad de tener programas que manejen la movilidad en el Terminal móvil (mecanismo utilizado por MIP), y descargar la responsabilidad de la movilidad de los usuarios enteramente a la red de acceso. El objetivo de NETLMM es establecer un estándar que funcione en todas las redes de acceso, ya que en la actualidad existen soluciones propietarias de conmutadores WLAN que permiten cierta movilidad pero que imponen límites de movilidad entre conmutadores de diferentes fabricantes.

La arquitectura de una red NETLMM se observa en la Figura 2.17. Estas redes están compuestas por LMA's (Local Mobility Anchor), MAG's (Mobile Access Gateway) y

MN's (Mobile Node). Un LMA enruta el tráfico hacia o desde los Nodos Móviles, mientras que un GMA es un elemento que cumple funciones de enlace frontera y gestiona la movilidad a nivel IP entre los enlaces frontera. Adicionalmente, el tráfico enviado hacia los nodos móviles por el LMA es enviado a través de un túnel entre el LMA y el MAG. Por otro lado, para gestionar la movilidad, NETLMM utiliza los mensajes IP que se requieren para realizar la conexión y desconexión de un Nodo Móvil (tales como los mensajes de Advertencia de los Encaminadores, mensajes de descubrimiento de grupos Multicast y mensajes de resolución de direcciones) junto con los mensajes de señalización de capa 2 para descubrir movimientos de los nodos móviles.

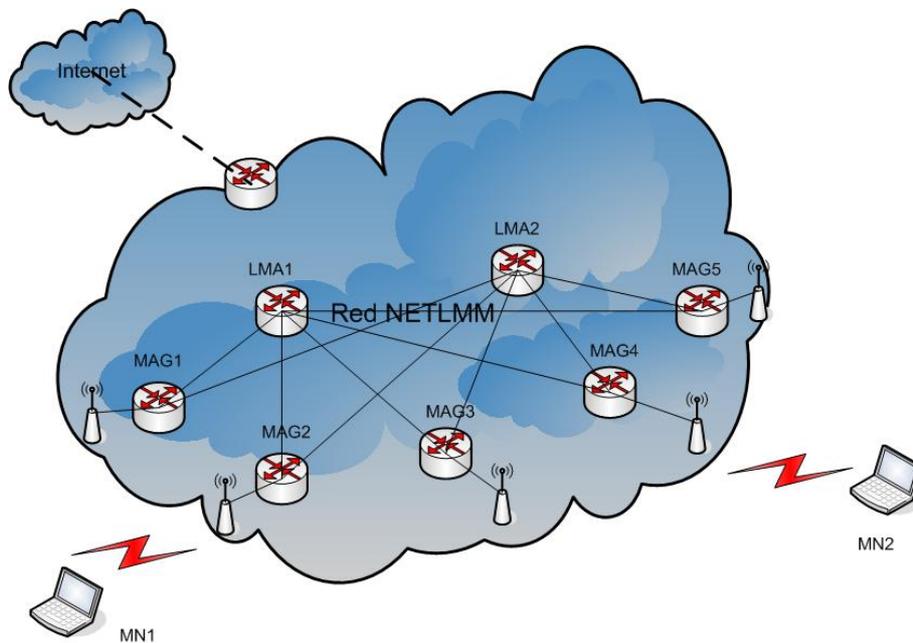


Figura 2.17. Red NETLMM

2.6 QoS en redes móviles todo IP

Las redes móviles aportan otro tipo de problemas a los ya establecidos en las redes fijas para garantizar la *Calidad de Servicio*. Su principal característica es que el *Sistema Final* es móvil y que por tanto puede moverse dentro de la subred e incluso entre subredes. Cuando un *Nodo Móvil* se mueve y realiza un *Traspaso*, el camino de los paquetes que van hacia el *MN* cambia, lo que provoca una interrupción en el flujo de los datos muy perjudicial para cualquier comunicación que esté realizándose [43]. El problema de *Calidad de Servicio* en redes móviles tiene especial interés de

investigación en la actualidad [44], pues hasta el momento se han hecho diferentes propuestas pero ninguna se ha adoptado como un estándar por la *IETF*.

2.6.1 Los problemas a solucionar

Aunque no existen soluciones perfectas, idealmente deberían solucionarse ciertos problemas para brindar *Calidad del Servicio* en redes móviles [44]. Tales problemas son:

➤ **Problemas de rendimiento**

- *Minimizar la interrupción de la QoS al momento de un Traspaso:* Los *Encaminadores* del nuevo camino podrían recibir paquetes del *MN* sin haberse enterado de sus requisitos de *QoS*. En ese caso los paquetes recibirían un tratamiento por defecto (aceptable al inicio de una sesión pero no al realizar un *Traspaso*).
- *Localizar el restablecimiento de QoS en las partes afectadas del camino de paquetes:* En caso de que el *Traspaso* cambie sólo un segmento próximo a un extremo del camino, el mecanismo de *QoS* debe limitar la extensión del restablecimiento de la *QoS* sólo al segmento afectado. Los protocolos de señalización de *QoS* tales como *RSVP* pueden localizar la señalización de *QoS* en las partes afectadas del camino extremo-extremo si la *CoA* no cambia durante el *Traspaso*. Si la *CoA* cambia, temporalmente puede haber doble aprovisionamiento de recursos (en el camino antiguo y el nuevo) hasta que finalice la temporización de refresco del estado de la reserva del camino antiguo.
- *Liberar el estado de QoS después del Traspaso a lo largo del camino de paquetes antiguo:* Es deseable que los estados de *QoS* que ya no se requieren a lo largo del camino antiguo, sean liberados lo más rápido posible al tiempo con el *Traspaso*. Puede suceder que el *MN* no siempre tenga la

oportunidad de enviar un mensaje explícito a lo largo del camino antiguo debido a la pérdida de conectividad de la capa de enlace con el *AR* antiguo.

➤ **Problemas de interoperatividad**

- *Con otros protocolos:* Existen protocolos complementarios a *IP Móvil* que están siendo definidos por la IETF o lo serán en un futuro. Algunos ejemplos son: *Fast Handoff*, Gestión de la movilidad localizada, Transferencia de contextos, etc. El esquema de *QoS* debe permitir complementar estas tareas aún si uno de estos protocolos o más no son usados.
- *Con caminos de paquetes que contienen diferentes mecanismos de QoS:* El nuevo camino después del *Traspaso* podría atravesar dominios de red que emplean diferentes mecanismos de *QoS* comparados con los del camino antiguo. El mecanismo de *QoS* debería ser capaz de establecer el tratamiento apropiado de *QoS* en adelante para los flujos de paquetes del *MN* a través de caminos de paquetes que usan diferentes mecanismos de *QoS* de una manera consistente.

➤ **Otros Problemas**

- *Soporte de QoS en diferentes tipos de caminos de paquetes:* Los caminos por los que podrían propagarse los paquetes después que el *MN* experimenta un *Traspaso* podrían ser variados: Camino de ruta optimizada entre *MN* y *CN*, *Triángulo de Encaminamiento* a través del *HA*, túnel temporal entre los *FA* antiguo y nuevo, túnel desde el nuevo *FA* al *HA*, túnel desde un *MAP* a un *MN*, etc. Un mecanismo de *QoS* debería ser capaz de soportar *QoS* a través de diferentes potenciales caminos de paquetes (esto es muy posible que no se pueda asegurar, ya que los caminos también dependen de otros mecanismos externos).

- *Interacción con el soporte de QoS de la capa de enlace inalámbrica:* El mecanismo de *QoS* para *IP Móvil* podría proveer alguna información a las capas de enlace inalámbrico para que ellas soporten la *QoS* requerida (p.ej. transmisión con diferente probabilidad de error). Un ejemplo de una tecnología de capa de enlace inalámbrica que soporta *QoS* es el estándar IEEE 802.11e [45].

2.6.2 Soluciones de Calidad del Servicio con IntServ

En [46] la solución que se plantea para extender los *Servicios Integrados* a los *MNs* consiste en incluir soporte explícito a la movilidad dentro de *RSVP*. La forma de incluir soporte a la movilidad en *RSVP* está basada en un método de “notificación” de movilidad, es decir, una notificación de la posición actual de un *MN* (Receptor) al Transmisor, realizada por el *HA* en una forma similar a la optimización de la ruta de *IP Móvil*. Este método implica un nuevo mensaje (*PATH Change*) de *RSVP* y una nueva clase de objeto (*Mobility Notification*) en *RSVP*. Esta solución tiene unas limitaciones bastante restrictivas, pues el *MN* no debe realizar un *Traspaso* y el *Nodo Correspondiente* debe ser un nodo fijo. Además, no ataca el problema de las reservas dentro de los túneles.

La propuesta planteada en [43], llamada también *MRSVP*, describe un protocolo de reserva para proveer servicios de tiempo real a usuarios móviles en una red de paquetes de *Servicios Integrados*. Para reducir el impacto de la movilidad en la *QoS*, un *MN* necesita hacer reservas de recursos por anticipado en ubicaciones múltiples que podría visitar posiblemente durante el tiempo de vida de la conexión. Esta aproximación es bastante drástica, pues propone un protocolo de reserva diferente a *RSVP*.

En [47], se propone realizar ciertas extensiones a *RSVP* para mejorar sus limitaciones en entornos móviles. Estas extensiones consisten en *Reservas de Recursos Inactivas*, *Receptores Virtuales* y *Pre-Asignación de Recursos Inteligente* basada en el análisis de los patrones de movilidad del usuario. Una idea básica de las extensiones propuestas es una expansión de la semántica de los mensajes de reserva *RSVP* incluyendo reservas

Comprometidas, Inactivas y Transientes. Otras ideas claves para las extensiones propuestas a *RSVP* incluyen *Receptores Virtuales* y *Pre-Reserva Inactiva de Recursos* a través de *Árboles Multicast Inactivos Dinámicos*. La combinación de características de *RSVP* como: el *Soft State*, orientación a receptor y reservas heterogéneas, mezcladas con las extensiones discutidas allí deberían incrementar la capacidad de *RSVP* para operar en ambientes móviles sin imponer cargas excesivas a los recursos de la red. Sin embargo, esta aproximación no plantea soluciones para las reservas dentro de los túneles y además exige grandes cambios al protocolo *RSVP*.

En la aproximación [48], se propone un protocolo alternativo a *IP Móvil* llamado *IP Móvil con Registros de Localización (MIP-LR, Mobile IP with Location Registers)*. Esta es una solución que plantea un cambio bastante radical con respecto a la solución planteada con *IP Móvil* para el problema de la movilidad.

La solución propuesta en [49], propone una nueva modificación al protocolo *RSVP* que es más flexible que *MRSVP*, ya que no requiere la especificación de los nodos a ser visitados. El protocolo propuesto intenta hacer *Re-Establecimiento Dinámico de Reservas* después de un *Traspaso*. Para ello utiliza un *Encaminador de Paso (CrossOver Router)*, que es un *Encaminador* que pertenece tanto al anterior camino del flujo como al nuevo camino del flujo después del *Traspaso* y que a su vez debe ser el más cercano al móvil. Esta propuesta intenta mejorar la movilidad de las reservas, aunque no ataca el problema de las reservas dentro de los túneles.

La propuesta descrita en [50], tiene las siguientes características:

- Este esquema usa versiones ampliadas de los protocolos *Mobile IPv4* y *RSVP* y permite reserva de recursos para flujos *Unicast* para *MNs*.
- Se utiliza *RSVP* para hacer señalización para reserva de recursos e *IP Móvil* para dar soporte a la movilidad.

- Se agrega una capa intermedia en lo alto de *IP Móvil* para permitir a *IP Móvil* informar a aplicaciones como *RSVP* los *Eventos de Movilidad*. Esto ofrece la ventaja de que otras aplicaciones pueden hacer uso de esta capa.
- Para dar solución al problema de reservas dentro de los túneles usa el *RFC 2746* [51].
- El túnel se establece entre el *HA*, que es el punto de entrada al túnel, y el *FA*, que es el punto de salida del túnel.

Otra propuesta está en [52] y utiliza *RSVP* para hacer reservas y *Mobile IPv6* para soportar movilidad. Esta propuesta plantea el problema de duplicación de reservas de recursos durante el *Traspaso* y trata de solucionarlo proponiendo una solución jerárquica con ciertas similitudes a *HMIPv6*. Propone el uso de un *Proxy* de movilidad *RSVP* (*RSVP Mobility Proxy*, o *RSVP-MP*), encargado de manejar los cambios de reservas durante el *Traspaso* dentro de un dominio conformado por *Encaminadores de Acceso Inalámbricos*, manteniendo inalterado el camino de la reserva en el resto de *Internet*. Para ello maneja dos tipos de direcciones, una *DCoA* (*Domain Care of Address*) y una *LCoA* (*Local Care of Address*), de tal forma que para el resto de *Internet*, se maneja la *DCoA*, mientras que para manejar las reservas dentro de su dominio usa la *LCoA*. Sin embargo, restringe los movimientos del *MN* dentro del Dominio del *RSVP-MP*.

2.6.3 Soluciones de Calidad de Servicio con DiffServ

En cuanto a soluciones con *DiffServ*, una de las propuestas estudiadas está descrita en [27]. Esta tiene una arquitectura de gestión de la movilidad basada en *Dominios de Registro* y supone que se usan protocolos como *Cellular IP*, *HAWAII*, o *TeleMIP* [5] para gestionar la movilidad y soportar *Fast Handoff*. Para adaptar *DiffServ* al ambiente móvil se hacen las siguientes consideraciones:

- Un *Dominio de Registro* equivale a un *Dominio DiffServ*.
- Un *Gateway Router* equivale a un *Encaminador de Borde* (*Edge Router*) que conecta con la red troncal de *Internet*.

-Las estaciones base equivalen a *Encaminadores de Borde* que proveen de acceso a *Internet* a los MNs.

Bajo estos supuestos, las funciones de un *Gateway Router* son: Interconectar hacia una red troncal de *Internet* de tipo *DiffServ*, negociar el *SLA (Service Level Agreement)* para especificar los recursos asignados por el proveedor de servicios de *Internet*, acondicionar el tráfico agregado por cada clase de servicio, asignar recursos y hacer el control de admisión de llamadas al *Dominio de Registro DiffServ*. Mientras que las funciones de las *Estaciones Base* son: Proveer a los *MNs* puntos de acceso a *Internet*, acondicionar el flujo de tráfico y realizar el marcado de los datos en el enlace ascendente.

En esta arquitectura se supone que todas las *Estaciones Base* pertenecientes a un mismo dominio, están conectadas al mismo *Gateway Router*. Además, los *Encaminadores DiffServ* usan tres colas separadas para proveer tres servicios: *Servicio Premium*, *Servicio Asegurado* y *Servicio Best-Effort*. Los tres *Buffers* son servidos bajo un planificador por prioridades o un planificador *WFQ (Weighted Fair Queuing)*. Así, las clases de tráfico de las redes 3G corresponden con:

- Conversacional (UMTS) – Premium (*DiffServ*)
- Streaming (UMTS) – Asegurado (*DiffServ*)
- Interactiva (UMTS) – Best-Effort (*DiffServ*)

Adicionalmente, existen algunas propuestas que utilizan una arquitectura *IntServ* sobre *DiffServ* y se detallan a continuación.

El trabajo en [53, 54] introduce tres mensajes *ICMP: Report, Reply* y *Modify*. Los mensajes *Report* llevan parámetros locales como ancho de banda o requisitos de potencia; los mensajes *Reply* confirman solicitudes de reserva. Los mensajes *Modify* están diseñados para cambios en ancho de banda. Este trabajo también utiliza colas basadas en clases y las extiende con la capacidad de reconocer la entrada *DSCP* en el campo *DS*. Esta propuesta además presenta extensiones para la pérdida de paquetes, cambios de asignación de clase y la capacidad de cambiar la prioridad de una cierta

clase. Más allá de la red de acceso, el trabajo asume que la red opera como una red *DiffServ* tradicional.

Finalmente, el trabajo en [55, 56] describe otra arquitectura *DiffServ* para usuarios de *IP Móvil*. Para movimientos *Intra-Dominio* en la *Red Doméstica*, el *MN* primero indica al *BB* (*Bandwidth Broker*) sus requisitos de *QoS* y la dirección *CoA* entre otras cosas. Después de revisar que hay suficientes recursos en la red, el *BB* reconfigura la red *DiffServ*. Si ambos procesos, la admisión y la reconfiguración, son satisfactorios, el *BB* envía un mensaje de confirmación al *MN*. Para movilidad inter-dominio, debe transferirse la especificación de nivel de servicio (*SLS*, *Service Level Specification*) del *MN*. Así, el *MN* primero solicita al *BB Visitado* que adquiera la *SLS* del *BB Local*. El *BB Local* envía entonces la *SLS* del *MN* y libera las reservas asociadas en la *Red Doméstica*. Entonces, el *BB Visitado* reconfigura su red e informa al *MN* ya sea del éxito o la falla del mensaje.

2.7 Alteraciones en el tratamiento de los paquetes IP en diferentes tipos de trayectos

Internet tiene diferentes situaciones que alteran el tratamiento normal de los paquetes IP en el trayecto desde su origen hasta su destino, pues impiden de alguna forma que se realice el procesamiento normal de las cabeceras de los paquetes. Dos situaciones de gran importancia son la *Agregación de Flujos* y la *Tunelización de Paquetes*. La primera, es utilizada entre otras situaciones en la red troncal de *Internet* para agilizar el Encaminamiento y disminuir el tamaño de las tablas utilizadas para este proceso. La *Tunelización* por su parte es una alteración que se presenta en diferentes situaciones tales como *VPNs* (*Virtual Private Networks*) y comunicación de un *Agente Propio* (*HA*) con un *Nodo Móvil* en *Protocolos de Movilidad en Internet*.

2.7.1 Encaminamiento con prefijos y Agregación de flujos en IPv6

Debido a que la *Agregación de Flujos* está estrechamente ligada al *Encaminamiento con Prefijos*, primero se describirá este tipo de *Encaminamiento* y luego se describirá cómo se realiza la *Agregación de Flujos*.

2.7.1.1 Encaminamiento con prefijos

Las tablas de Encaminamiento tradicionales de *IPv4* contienen una entrada por cada red, definida por la *Clase de Red*, y una entrada por cada subred bajo una red determinada. En otras palabras, cada red alcanzable debe tener una entrada separada en la tabla de Encaminamiento. Los protocolos de Encaminamiento antiguos no intercambian información de prefijos, y así requieren que las decisiones de Encaminamiento sean hechas basadas en la clase de las direcciones para redes remotas, o en la máscara de subred para las redes conectadas directamente. Esta opción también requiere que la misma máscara de subred, o prefijo de red sean usadas en todos los sitios en la red y que todas las subredes sean alcanzables sin transitar hacia otra red [57].

Por otro lado, los nuevos protocolos de Encaminamiento intercambian información de los prefijos, permitiendo el uso de máscaras de subred de longitud variable (*VLSM-Variable Length Sub-Net Mask*) [58] en diferentes subredes y además el uso de agregación arbitraria de números de redes y subredes dentro del más pequeño número de rutas posible [57]. El intercambio de información de prefijos *VLSM*, el cual es adoptado en *IPv6*, permite tablas de Encaminamiento pequeñas y un uso más eficiente del espacio de direccionamiento. El mecanismo de agregación de rutas basado en el prefijo está basado en el Encaminamiento Inter-dominio sin clases (*CIDR- Classless Inter Domain Routing*) [58, 59] usado en *IPv4*.

Para mantener las tablas de Encaminamiento bajo control y ayudar a asegurar la eficiencia del Encaminamiento basado en prefijos, deben asignarse direcciones a los proveedores de acceso de red (*Network Access Providers*). Inmediatamente surgen una serie de problemas; el primero de ellos es que un cambio en el proveedor resulta en que el sitio entero debe ser re-enumerado. Para facilitar el problema administrativo de re-

enumeración de la dirección del sitio, las direcciones en *IPv6* pueden expirar y ser generadas automáticamente. Sin embargo, la expiración de las direcciones causa problemas con conexiones de larga vida, ya que la conexión sigue viva pero las direcciones finales han cambiado. Este problema es resuelto parcialmente a través del uso de tiempos de vida asociados con cada dirección. El primer tiempo de vida es el “Tiempo de vida válido”, que define qué tanto tiempo es legal el uso de la dirección. El segundo tiempo de vida es el “Tiempo de vida preferido”, que es usado para definir cuánto tiempo debería ser usada la dirección antes de que deba obtenerse y usarse una nueva dirección. Si una conexión de larga vida desea persistir después de la expiración de la dirección, deben usarse mecanismos de la capa de transporte para preservar la dirección después del cambio de dirección.

2.7.1.2 Agregación de flujos

La *Agregación* [59] es realizada para reducir los requisitos de desempeño en los *Encaminadores* y para ayudar al Encaminamiento con prefijos. El formato propuesto para direcciones *Unicast* globales agregables basadas en el proveedor, se muestra en la Figura 2.18.

FP	TLA ID	NLA ID	SLA ID	Interface ID
3	13	32	16	64

Figura 2.18 Formato de dirección *Unicast* global agregable de *IPv6*

El campo *Prefijo de Formato* (*FP- Format Prefix*) indica qué formato de dirección está siendo usado. Los bloques de direcciones serán asignados a proveedores de tránsito en el rango disponible en el campo *TLA ID* (*Top-Level Aggregation Identifier*). Los proveedores de tránsito del nivel más alto, que manipulan la red *Núcleo*, asignarán entonces bloques de direcciones en el campo *NLA ID* (*Next-level Aggregation Identifier*) para identificar las redes de tránsito y los sitios servidos por el proveedor. Cada sitio podría asignar direcciones en el campo *SLA ID* (*Site Level Aggregation Identifier*) a redes individuales dentro del sitio. El campo *Identificador de Interfaz*

(*Interface ID*) está basado en la dirección de 64 bits *IEEE EUI-64* del *Hardware* conectado a la red. De esta manera las direcciones de *Interfaz*, o direcciones de capa 2, proveen un mecanismo directo para generar automáticamente una dirección *IPv6* única globalmente.

El uso de prefijos en el *Direccionamiento Agregado* permite un Encaminamiento simplificado y eficiente. El *Encaminamiento con Prefijos* usa un prefijo de longitud variable para crear jerarquías con *Redes de Tránsito* y *Redes Hoja (Redes de Acceso)*[57]. Para reducir el problema de re-enumeramiento de sitios, los prefijos podrían expirar. El prefijo son los primeros n bits de la dirección, donde n es la longitud del prefijo. Usando la notación de prefijos estándar, el prefijo $3FFE:0900::0/24$ podría significar que los primeros 24 bits de la dirección, es decir $3FFE:09$ son el prefijo. Para direcciones *Unicast*, el prefijo también identifica una dirección específica, si el prefijo es de 128 bits de longitud, o una *Red de Tránsito*, si el prefijo es de 1 a 64 bits de longitud. Los prefijos para otros tipos de direcciones podrían tener otro significado. Los prefijos podrían indicar si son *On-Link* u *Off-Link*. Un prefijo *On-Link* significa que los nodos con ese prefijo son alcanzables en una red conectada directamente. Un prefijo *Off-Link* significa que debe usarse una pasarela (*Gateway*) para alcanzar nodos usando este prefijo. La clasificación *On-Link* y *Off-Link* simplifica las decisiones de Encaminamiento local y de re-enumeración. Esta clasificación es transportada por el protocolo de descubrimiento de red.

En la Figura 2.19 se observa un ejemplo de *Encaminamiento con Prefijo*. El *Prefijo 1* es de 32 bits y cubre las redes en sentido *Downstream* y los nodos servidos por prefijos 2, 3 y 4, asumiendo que no hay nodos “*Multi-Domiciliados*” y así sucesivamente. Las redes en sentido *Downstream* son cubiertas por el *Prefijo 1* ya que está contenido en los prefijos y direcciones de cada nodo que está conectado a la *Red de Tránsito* en sentido *Downstream* del ejemplo. El *Prefijo 2* de 48 bits identifica un sitio alcanzable por la *Red de Tránsito* provista por el nodo que usa el *Prefijo 1*. Los prefijos 3 y 4 de 64 bits son *Redes Hoja* que son alcanzables por la *Red de Tránsito* conectada bajo el prefijo 2. Así, para los anuncios de ruta en sentido *Upstream* o de tipo *Peer* (igual o par), los *Encaminadores* en la red que usan el *Prefijo 1* sólo necesitan advertir el *Prefijo 1*. Todos los nodos y redes que están cubiertas por los prefijos 2, 3 y 4 son tomados en

cuenta por las advertencias para el *Prefijo 1*. Supongamos que un *Encaminador* usando el *Prefijo 4* es un par de un *Encaminador X*, el cual también es par del *Encaminador* que usa el *Prefijo 1*. El *Encaminador X* recibirá advertencias para los prefijos 1 y 4, y deberá seleccionar los destinos usando el prefijo de comparación más largo. Por ejemplo, si un paquete está vinculado al nodo 3FFE:900:1:111:260:97FF:FE6C:57BF, el *Encaminador X* podría enviar paquetes hacia el *Encaminador* que sirve el *Prefijo 1*. Si un paquete está vinculado al nodo 3FFE:900:1:112:2A0:24FF:FED0:C6B4, el *Encaminador X* podría enviar el paquete hacia el *Encaminador* que sirve el *Prefijo 4*.

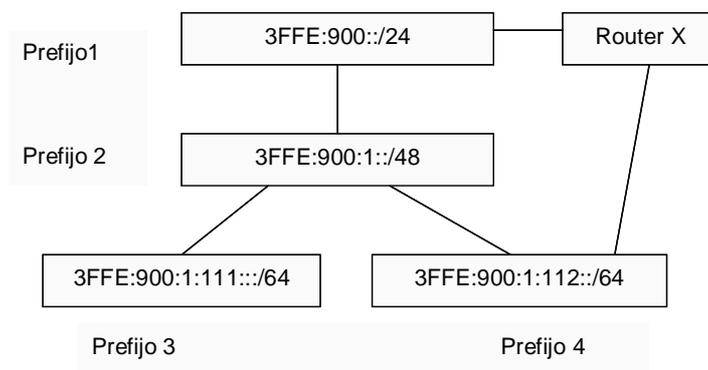


Figura 2.19 Ejemplo de Encaminamiento con prefijo.

2.7.2 *Tunelización de paquetes genérica en IPv6*

El RFC 2473 [42] especifica un método por el cual un paquete es encapsulado y transportado como carga útil dentro de un paquete IPv6. El paquete resultante es llamado *Paquete de Túnel IPv6*.

IPv6 Tunneling es una técnica para establecer un *Enlace Virtual* entre dos nodos IPv6 para transmitir paquetes de datos como carga útil de paquetes IPv6. El RFC 2473 [42] se refiere en particular a túneles entre dos nodos identificados por direcciones *Unicast* (similar a un enlace punto-punto). Los mecanismos descritos se aplican también a túneles cuyo punto de salida está identificado por direcciones *Anycast* y *Multicast* (similar a un enlace punto-multipunto). Un túnel se establece en una sola dirección origen-destino, por lo que si se requiere comunicación en los dos sentidos entre el

origen y el destino, deberá establecerse un túnel para cada sentido de la comunicación tal como se observa en la Figura 2.20.

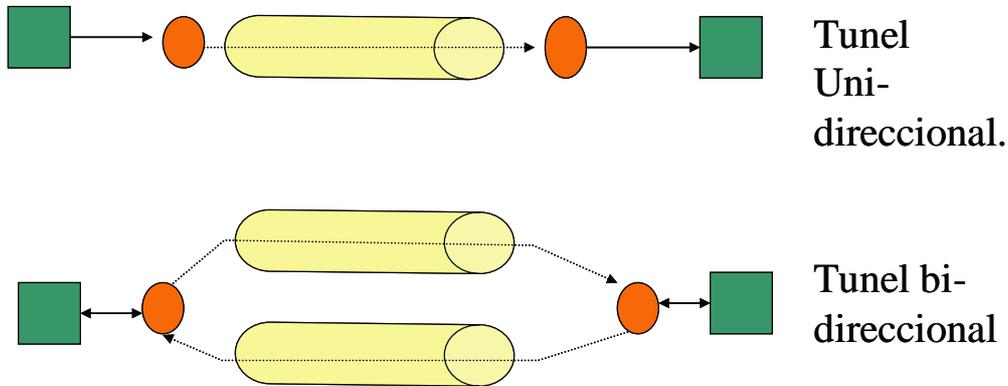


Figura 2.20 Tipos de túneles en *IPv6 Tunneling*

➤ *Encapsulado IPv6*

El *Encapsulado IPv6* consiste en agregar al paquete original una cabecera *IPv6* y, opcionalmente, un conjunto de cabeceras de extensión *IPv6*, que son llamadas en conjunto *Cabeceras IPv6 de Túnel*. Esto sucede en el punto de entrada al túnel tal como se observa en la Figura 2.21.

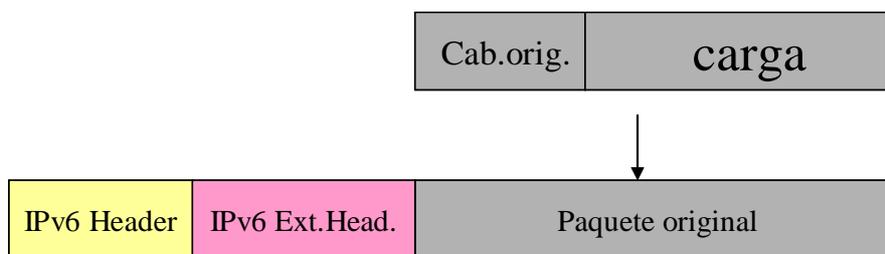


Figura 2.21 Encapsulado *IPv6*

El paquete original es procesado durante el envío de acuerdo a las reglas de envío del protocolo de ese paquete. Si el paquete es *IPv6*, el límite de saltos de la cabecera *IPv6* original es decrementado en 1. Por otro lado, si el paquete es *IPv4*, el campo *TTL* de la cabecera original es decrementado en 1.

Durante el encapsulado, el campo *Fuente* de la cabecera *IPv6* del túnel se llena con la dirección *IPv6* del nodo de entrada al túnel y el campo *Destino* con la dirección *IPv6* del nodo de salida del túnel.

➤ *Procesamiento de los paquetes en los túneles*

Los nodos intermedios en el túnel *IPv6* procesan los paquetes del túnel de acuerdo al protocolo *IPv6*. Así, de acuerdo al tipo de cabecera de extensión, el procesamiento de dicha cabecera será diferente:

- *Cabecera de Opciones Hop by Hop*: Es procesada por cada nodo en el túnel.
- *Cabecera de Encaminamiento*: Identifica los nodos de procesamiento intermedios y controla con una granularidad fina el camino de los paquetes del túnel a través de él.
- *Cabecera de opciones del Destino*: Es procesada en el nodo de salida del túnel.

➤ *Des-encapsulado IPv6*

Después de recibir un paquete *IPv6* en el nodo de salida del túnel, su capa de protocolo *IPv6* procesa las cabeceras de túnel. Esto se hace en estricto orden de izquierda a derecha. Cuando el procesamiento se completa, el control es pasado al siguiente protocolo (ver Figura 2.22), el cual es identificado por el valor del campo *Next Header* en la última cabecera procesada. Si este campo tiene el valor *Protocolo de Túnel*, el *Protocolo del Túnel* descarta las cabeceras de túnel y pasa el paquete original a la capa de *Internet* o el protocolo identificado por ese valor para su procesamiento.

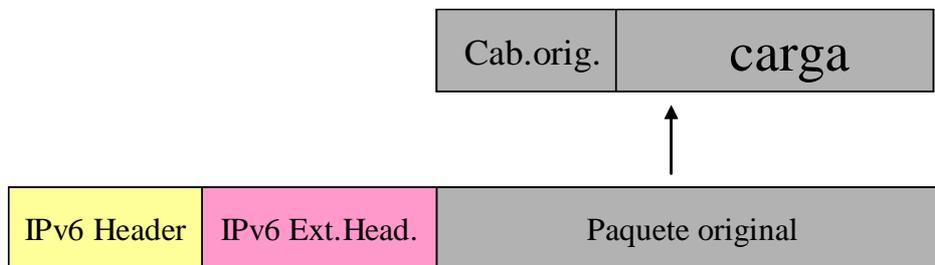


Figura 2.22 Des-encapsulado *IPv6*

➤ Encapsulación anidada

Esta situación consiste en que se construye un túnel dentro de otro túnel (ver Figura 2.23). Esto podría conllevar a un gran número de túneles anidados, lo que podría traer algunos inconvenientes si el número de anidamientos es muy grande debido a que si hay fragmentación de un paquete a la entrada de un túnel interior de un paquete de túnel ya fragmentado, resulta en que se dobla el número de fragmentos. Por tanto, se debe evitar que se haga un número de anidamientos excesivo transportando una opción llamada *Límite de Encapsulado del Túnel* en una cabecera de extensión llamada *Cabecera de Opciones del Destino IPv6*.

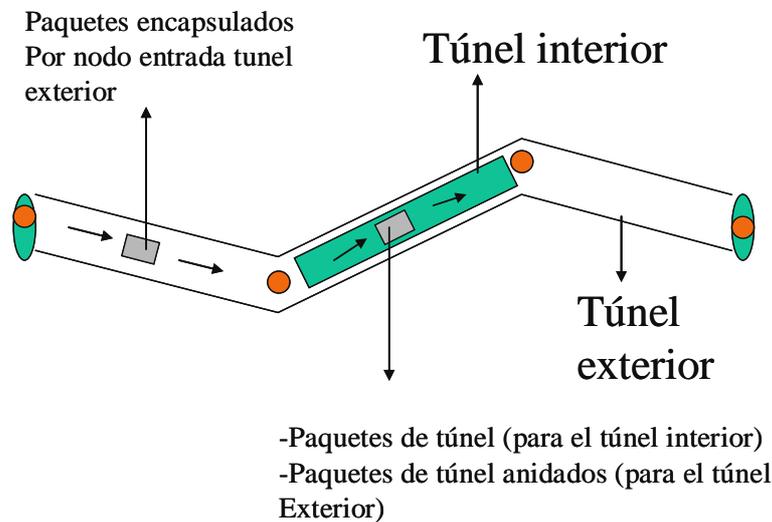


Figura 2.23 Encapsulación anidada

3 IntServ6: Propuesta para el soporte de Calidad de Servicio en Internet

Las redes de próxima generación estarán basadas en el protocolo *IPv6* [18]. Como resultado, estas redes tendrán características que el protocolo *IPv4* no puede soportar tales como un gran número de direcciones IP, *Calidad de Servicio*, Seguridad, etc. Tales redes tendrán que soportar diferentes clases de servicios para aplicaciones elásticas y aplicaciones de tiempo real. Bajo este ambiente, una característica esencial de este tipo de redes será el soporte de *Calidad de Servicio*.

En la actualidad, el soporte de *Calidad de Servicio* está basado principalmente en dos arquitecturas estándar: La *Arquitectura de Servicios Integrados (IntServ- Integrated Services Architecture)* [19] y la *Arquitectura de Servicios Diferenciados (DiffServ- Differentiated Services Architecture)*[20]. La *Arquitectura de Servicios Integrados* es utilizada principalmente en *Redes de Acceso* debido a que se adapta fácilmente a las necesidades de recursos de los usuarios, pero a su vez tiene problemas de escalabilidad debido al agotamiento de los recursos de la red. Por otro lado, la *Arquitectura de Servicios Diferenciados* es muy escalable y por tanto, soporta una gran cantidad de usuarios, pero a cambio, no puede ajustarse de forma precisa a las necesidades de recursos de los usuarios. Por tanto, *DiffServ* es utilizada principalmente en *Redes de Transporte*. Un componente adicional dentro de este escenario son las redes *MPLS (Multi-Protocol Label Switching)* [21] que soportan los principios de Ingeniería de tráfico. Adicionalmente, *MPLS* puede complementarse con la *Arquitectura de Servicios Integrados* o con la *Arquitectura de Servicios Diferenciados* para soportar *QoS* de una mejor manera en una *Internet*.

La propuesta planteada en esta tesis, llamada *IntServ6* [6], está basada en la *Arquitectura de Servicios Integrados* pero con una diferencia importante en el proceso de clasificación de paquetes: mientras la *Arquitectura de Servicios Integrados* estándar (*IntServ*) utiliza la *Quíntupla*² para la clasificación de los paquetes, *IntServ6* utiliza el

² La *Quíntupla* es el conjunto de campos compuesto por las direcciones IP de origen y destino, los puertos de origen y destino y la identificación del protocolo.

campo de *Etiqueta de Flujo* para identificar el flujo al que pertenece un paquete. Esta idea no es nueva y además, es la motivación original por la cual se creó el campo de *Etiqueta de Flujo* dentro de la cabecera *IPv6* [18]. Sin embargo, hasta ahora este campo no ha sido usado para este propósito. Además, la utilización del campo de *Etiqueta de Flujo* no ha sido estandarizada completamente. Aunque los estándares actuales [60, 61] describen recomendaciones generales, ellos no son claros acerca del uso específico de este campo. En *IntServ6*, el uso del campo de *Etiqueta de Flujo* para clasificar los paquetes mejora el desempeño de los *Encaminadores* y mejora otras características en las que la aproximación *IntServ* estandarizada presenta deficiencias. Estas características incluyen la agregación de flujos, las reservas dentro de los túneles y la interconexión con redes de transporte. Adicionalmente, *IntServ6* reduce la dependencia del retardo de los paquetes con respecto a la movilidad de los usuarios, lo que permite a *IntServ6* ofrecer una *QoS* más robusta en redes móviles que el estándar *IntServ*.

Este capítulo describe la propuesta *IntServ6* y sus diferentes aspectos. Para ello, en la sección 3.1 se hace una descripción del vacío en la estandarización y se describen algunas propuestas sobre el uso del campo de *Etiqueta de Flujo* en *IPv6*. Luego, la sección 3.2 describe el proceso de *clasificación de paquetes*, el cual es un aspecto fundamental de *IntServ6*. Posteriormente, la sección 3.3 se detalla el procedimiento para el soporte de reservas en túneles. La sección 3.4 describe cómo se realiza la *Agregación de flujos*, la sección 3.5 explica cómo se realiza la interconexión de redes *IntServ6* con redes *MPLS*, la sección 3.6 describe las extensiones para la movilidad en *IntServ6* y finalmente, la sección 3.7 verifica el cumplimiento de las especificaciones de la *Etiqueta de Flujo* que han sido descritas en los estándares actuales.

3.1 El vacío en el conocimiento: Uso de la Etiqueta de Flujo de IPv6 para el soporte de QoS.

La integración de *IPv6* y las tecnologías de *Conmutación de Etiquetas (Label Switching)* ya ha sido estudiada en el pasado. En 1996, Baker y Rekhter, publican un *Draft* (Borrador) de la *IETF (Internet Engineering Task Force)* [62]. Por esta época se debatió en el *IPng Working Group* de la *IETF* pero la propuesta no prosperó. Sin

embargo, en 1997 Rekhter publica el *RFC* 2105 [63]. Dicho *RFC*³ especifica que se debe usar la *Etiqueta de Flujo IPv6*, pero no describe cómo sería la semántica. A continuación se produjo una dura lucha por establecer una solución de *Conmutación de Etiquetas* y surgieron soluciones como *Tag Switching* de *CISCO*, *Aggregate Route-based IP Switching (ARIS)* de *IBM*, *IP Navigator*, *IP Switching de IPsilon*, *Cell Switch Router (CSR)*, y finalmente, la propuesta que llegó a ser un estándar de la *IETF* y que es la solución para muchos de los problemas de *Internet: MPLS (Multiprotocol Label Switching)* [21]. Sin embargo, la estandarización del uso del campo de la *Etiqueta de Flujo* de la cabecera *IPv6* sigue sin ser establecida. Los estándares de la *IETF* tales como el *RFC* 2460 [18] que describe *IPv6* y el *RFC* 3697 [61] que detalla la especificación de flujos *IPv6*, sólo mencionan aspectos generales y ciertas recomendaciones acerca de la utilización de la *Etiqueta de Flujo* pero no son precisos acerca de la forma exacta en que se debe utilizar tal campo. Es así, como han surgido diferentes propuestas de las cuales algunas se mencionan a continuación.

➤ *IP Next Generation Label Switching*

Esta aproximación se encuentra en [64] y propone el re-envío de paquetes *IPv6* usando técnicas de conmutación de etiquetas con las mismas ventajas de la arquitectura *MPLS*. Dicho documento propone una correspondencia de todos los campos de la cabecera *MPLS* dentro de la cabecera *IPv6*, obteniendo varias ventajas como la simplicidad del modelo y el decremento de la sobrecarga. Esta correspondencia es llamada *IP Next Generation Label Switching*, o sólo *IPngLS* y puede trabajar concurrentemente con *MPLS*, es decir, los paquetes *IPv6* son retransmitidos mediante *IPngLS* y los paquetes *IPv4* son retransmitidos mediante *MPLS*. Para probar *IPngLS* con redes del mundo real, se desarrolló una implementación dentro del *Kernel* del sistema operativo *Linux*. Esta propuesta se centra exclusivamente en la correspondencia de los campos de la cabecera *MPLS* dentro de la cabecera *IPv6* y no soluciona otros aspectos.

³ *RFC* es la sigla de *Request for Comments*. Esta sigla se utiliza para designar los estándares de *Internet* generados por la *IETF*, que es la entidad encargada de estudiar y crear tales normas.

➤ *Signaling in IP Cell-Switching*

Esta aproximación, descrita en [65], presenta una propuesta para hacer conmutación de celdas *IP* denominada *IP Cell-Switching*. Esta solución está basada en un paquete *IPv6* de longitud fija y en la tecnología de *Conmutación de Etiquetas* con diferentes calidades de servicio. La etiqueta utilizada es el campo de *Etiqueta de Flujo* de la cabecera *IPv6*. En este documento se indica que debe establecerse un camino virtual antes de enviar el flujo de celdas *IP*, por lo que se requiere un protocolo de señalización para establecer las reservas de recursos necesarias para asegurar la *Calidad de Servicio* solicitada por el usuario.

La propuesta analiza los protocolos de reserva existentes que usan un camino por defecto. Este protocolo de señalización debe ser apropiado para elegir un camino que provea la *QoS* negociada entre el transmisor y el receptor antes de reservar algunos recursos. La mayoría de los protocolos existentes determinan el mejor camino en el momento del establecimiento. Sin embargo, el mejor camino podría congestionarse un tiempo después, por lo que, dividir el flujo sobre múltiples caminos podría dar un mejor desempeño que usar sólo un camino. Finalmente, se concluye que el protocolo *RSVP* podría trabajar como un protocolo de señalización para soportar un esquema de reserva de recursos en un ambiente de conmutación de etiquetas. Además, este ambiente está bien adaptado al paradigma de *multicast*. *IP Cell-Switching* es una propuesta bastante antigua que trata de justificar el uso de *RSVP* para establecer reservas. Además, se fundamenta en el uso de paquetes *IPv6* de longitud fija.

➤ *QoS Provisioning Using IPv6 Flow Label in the Internet*

Esta aproximación [66] se enfoca en el soporte de *QoS* usando el campo de *Etiqueta de Flujo* de *IPv6*. Primero se investiga la especificación de la *Etiqueta de Flujo* y su uso para el soporte de *QoS* en la literatura, luego se propone un esquema de *QoS* extremo a extremo (*End-to-End*) usando la *Etiqueta de Flujo* de *IPv6*. Esta propuesta emplea una aproximación híbrida para la especificación de la *Etiqueta de Flujo* y una *Opción de Etiqueta de Flujo* para paquetes *IPv4*. Este esquema de *QoS* propuesto está especialmente diseñado para una red en transición que contiene nodos de red de ambos

tipos, *IPv4* e *IPv6*, lo que es una realidad en el proceso de transición de *IPv4* a *IPv6*. Los resultados de simulación muestran que el esquema de *QoS* propuesto puede diferenciar eficientemente varios tipos de tráfico para obtener los requisitos de *QoS* diferenciados en una red en transición que contiene *Encaminadores IPv4* e *IPv6*. En esta solución, los autores se centran en la descripción de dos formatos para la *Etiqueta de Flujo*, pero no describen cómo debe ser el funcionamiento de los *Encaminadores*.

- *Implementing QoS capabilities in IPv6 networks and comparison with MPLS and RSVP*

En esta propuesta [67], se investiga una metodología para el soporte de *QoS* extremo a extremo basada en *IPv6* y se estudian parámetros como el retardo extremo a extremo, las pérdidas de paquetes y el comportamiento del retardo usando simulaciones desarrolladas con *Network Simulator*. Los resultados obtenidos son comparados con dos métodos de *QoS* extremo a extremo: *MPLS* y *RSVP*. En [67] no se presenta una descripción del funcionamiento de los *Encaminadores*.

- *Comparison of QoS performance between IPv6 QoS management model and IntServ and DiffServ QoS models*

En este artículo [68] se comparan tres esquemas de *QoS*: *Servicios Integrados*, *Servicios Diferenciados* y *Gestión de QoS de IPv6 (IPv6 QoS Management)*. Esta comparación se realiza con respecto a requisitos de garantía de *QoS*. Además se discuten características como la complejidad y la escalabilidad para cada aproximación.

El modelo *Gestión de QoS de IPv6* se desempeña bien durante la simulación. También es simple y escalable cuando se implementa. El esquema propuesto incorpora los elementos necesarios para manejar solicitudes de *QoS* mediante el procesamiento, monitorización y control de flujos de tráfico. Se usa *WFQ* para separar los flujos en colas de tráfico separadas para cada flujo. El sistema *Gestión de QoS de IPv6* usa la combinación de la *Etiqueta de Flujo* y la *Dirección IP Origen* (a lo que se le llama *Domain Global Identifier, DGI*) para reservar y seguir la pista de los flujos de tráfico. El campo *Clase de Tráfico (TC)* es usado para la planificación de paquetes en las colas

de los *Encaminadores*. Este esquema permite que la red *IPv6* pueda gestionar la *QoS* sin necesidad de invocar otros protocolos de *QoS* tales como *RSVP* y *MPLS*. El gestor de *QoS* usa el *DGI* para definir de forma única los flujos de tráfico para seguirlos y reservar recursos.

El retardo extremo a extremo es menor en comparación con los *Encaminadores IP* ya que los *Encaminadores* del *Núcleo* usan el *DGI* para las decisiones de retransmisión de paquetes en lugar de usar la *Quíntupla*. Las desventajas de *IntServ* y *DiffServ* son resueltas en esta técnica de *QoS* así:

- Los *Encaminadores Frontera* manejan solicitudes de *QoS* y se comunican con otros gestores de *QoS*.
- Los nodos que generan paquetes no negocian solicitudes con el gestor de *QoS* como en el caso del *BB (Bandwidth Broker)* en un Dominio *DiffServ*.
- Los flujos de tráfico son clasificados con base en el campo *TC* por lo que cada nivel de prioridad es tratado de forma diferente. Por tanto, se evita la limitación de clases y no se requiere mapear los flujos de paquetes a clases pre-definidas (como en *DiffServ*), lo que resultaría en una capacidad de manejo de *QoS* limitada.
- Este modelo es simple de implementar ya que no requiere de algoritmos de señalización o admisión.

En esta propuesta (ver Figura 3.1), cuando un nodo generador quiere enviar tráfico en tiempo real, envía una solicitud de *QoS* al *Encaminador Frontera* de la red. Después de recibir la solicitud del transmisor, el *Encaminador Frontera* se comunica con su *Gestor de Dominio* para aprobar o rechazar estas solicitudes. El *Encaminador Frontera* retransmite las respuestas del *Gestor*, sean positivas o negativas, hacia el transmisor. Si la solicitud es aceptada, la fuente inicia el envío de paquetes hacia el *Encaminador Frontera* donde los paquetes de los flujos de tráfico son clasificados, planificados y monitoreados. Los paquetes son puestos en cola dependiendo de su prioridad y las políticas establecidas por el *Gestor de QoS*. El esquema de *QoS IPv6* usa el algoritmo de *Token Bucket* para que el tráfico aceptado cumpla con las especificaciones de tráfico.

Cuando la tasa de un flujo se incrementa y excede su especificación solicitada, su nivel de prioridad es degradado o sus paquetes son desechados.

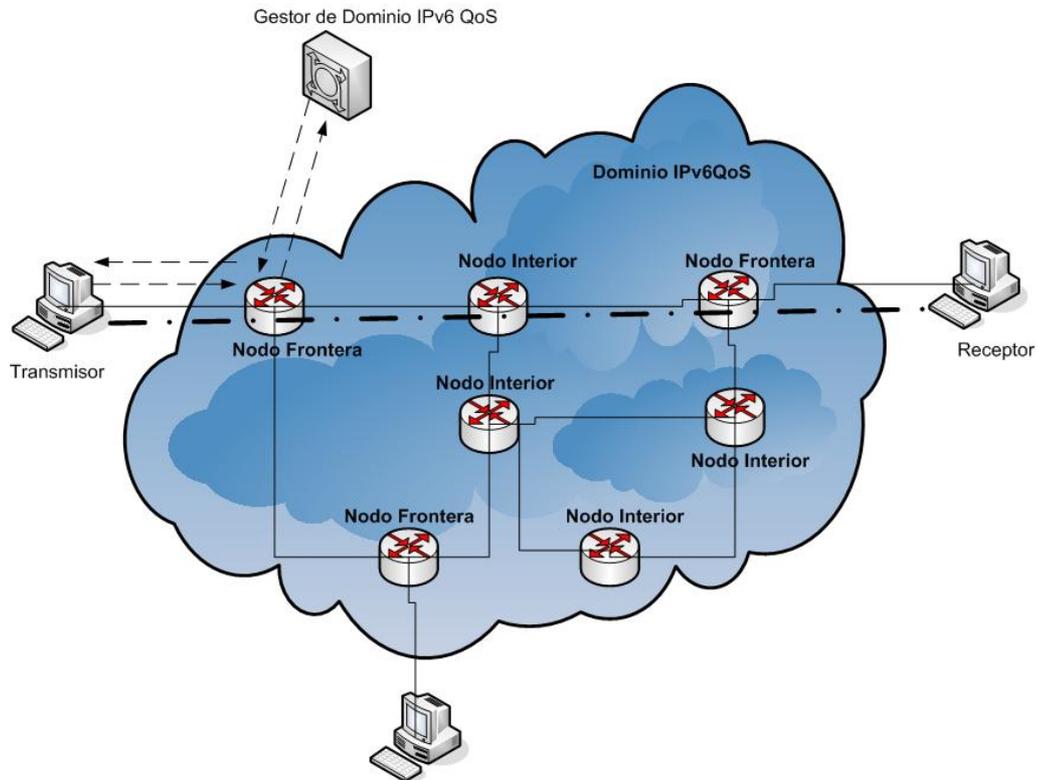


Figura 3.1 Operación del esquema de *QoS IPv6*.

En conclusión, el esquema de *gestión de QoS IPv6* es simple y menos complejo que *IntServ*, ya que sólo el *Encaminador Frontera* se involucra en la admisión, monitoreo y suavizado, mientras que los *Encaminadores* interiores sólo retransmiten y planifican. Además, los paquetes son procesados más rápido ya que se usa el *GDI* en lugar del Encaminamiento *IP Lookup* normal, el cual involucra los campos de la *Quíntupla* de la cabecera.

La propuesta es un híbrido entre *DiffServ* e *IntServ* y no está claro cómo se hacen las solicitudes de *QoS* a los *Encaminadores Frontera*. Usa una combinación de la *Etiqueta de Flujo* y de la *Dirección IP Origen* para hacer la clasificación pero después dice que para clasificar los flujos (realmente es para hacer la planificación) se utiliza el campo *Clase de Tráfico (TC)*, por lo que no es claro sobre cuál es la política de clasificación.

En conclusión, esta propuesta habla de manejar reservas mediante el *DGI* pero luego en la planificación usa sólo el *TC*, lo que es contradictorio. Esto requeriría que todos los *Encaminadores Frontera* tuviesen la información acerca de todos los recursos de la red para saber si hay suficientes recursos en ella para aceptar un nuevo flujo ya que una vez se acepta, dentro de la red no se tiene en cuenta la reserva en cada *Encaminador* sino la prioridad del flujo como en *DiffServ*. Entonces se requeriría un protocolo de señalización para mantener la información de reservas actualizada en todos los *Encaminadores* de la red.

3.2 Clasificación de flujos en IntServ6

El fundamento principal de *IntServ6* es el uso de una *Arquitectura de Servicios Integrados* para soportar *Calidad de servicio* en redes *IPv6*. Además, se busca que el desempeño de los *Encaminadores* de *Internet* que soportan *Servicios Integrados* sea mayor que el que se ofrece actualmente. Para ello *IntServ6* utiliza el campo *Etiqueta de Flujo* de la cabecera *IPv6* para agilizar el proceso de *Clasificación de Paquetes* en los *Encaminadores*. Este proceso es realizado dentro del modulo de *Identificación de Flujos* y es un aspecto clave para diferenciar nuestra propuesta con respecto a el estándar *IntServ*.

3.2.1 Clasificación de paquetes con IntServ

El módulo de *Identificación de Flujos* (también llamado *Clasificador*) debe examinar cada paquete entrante y decidir si el paquete pertenece a uno de los flujos *RSVP* reservados. La identificación de un flujo *IP* se hace mediante 5 campos de la cabecera del paquete (también conocidos como *Quíntupla*): *Dirección IP Origen*, *Dirección IP Destino*, *Identificación del Protocolo*, *Puerto Fuente*, *Puerto Destino*. Para determinar si un paquete coincide con un flujo *RSVP*, el motor de *Identificación de Flujos* debe comparar la *Quíntupla* del paquete entrante con la *Quíntupla* de todos los flujos en la *Tabla de Reservas*, si hay una coincidencia, se obtiene el estado de la reserva a partir de la *Tabla de Reservas* y el paquete es enviado al *Planificador de Paquetes* junto con el estado de la reserva asociado con el flujo. Este proceso debe hacerse en tiempos muy

cortos pues la cantidad de paquetes de un enlace entrante de un *Encaminador* puede ser muy grande, al igual que el número de flujos.

Para solucionar el problema de la eficiencia en la clasificación de los paquetes, *IntServ* utiliza una solución denominada *Hashing*. Esta consiste en que, durante el establecimiento de la *Reserva*, el *Encaminador* calcula un valor a partir de una *Función Hash* que utiliza la *Quíntupla* como entrada. El valor *Hash* de salida es usado para identificar el estado de la reserva del flujo en la *Tabla de Reservas* del *Encaminador* (ver Figura 3.2) y se almacena en una *Tabla Hash* que contiene el valor *Hash* y un apuntador a la *Tabla de Reservas*. Si el valor *Hash* calculado ya existe en la tabla se dice que hay una *Colisión* y se coloca para esta entrada de la tabla el *Bit de Colisión* en 1, indicando que hay una colisión. En este caso se almacena en la *Tabla de Resolución de Colisiones* el valor *Hash* y la *Quíntupla* correspondiente, junto con un apuntador a la reserva respectiva en la *Tabla de Reservas*.

Durante el proceso de retransmisión de paquetes, el *Encaminador* obtiene el valor *Hash* a partir de la *Quíntupla* de cada paquete entrante y luego verifica en la *Tabla Hash* si este valor existe. Si no existe, el paquete es tratado con un servicio tipo *Best-Effort*, mientras que si el número *Hash* existe en la tabla, se debe verificar si hay una *colisión*. En caso de no haber *colisión*, el número *Hash* apunta directamente a la *Tabla de reservas*, mientras que, si hay *colisión*, se debe resolver esta haciendo una búsqueda del número *Hash* y de la *Quíntupla* en la *Tabla de resolución de colisiones*. Este proceso de clasificación es un caso particular de la clasificación tipo *Matching* (ver sección 2.3.2).

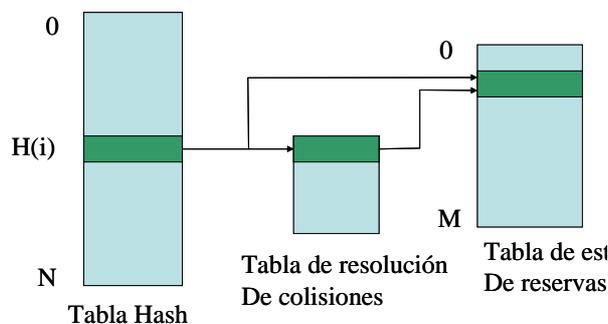


Figura 3.2 Procedimiento de *Hashing* en *IntServ*.

3.2.2 Clasificación con IntServ6

IntServ6 agiliza el proceso de *clasificación de paquetes* en los *Encaminadores IntServ* para el caso de *IPv6* [6-8]. Para ello, se descarga la responsabilidad del cálculo del *Número Hash* en el *Host* origen del flujo. Esto puede hacerse sin problema ya que mientras el algoritmo de cálculo sea el mismo siempre, si la semilla es la misma (la *Quíntupla*), no importa el lugar donde se aplique la función *Hash*, siempre arrojará el mismo resultado. El número *Hash* es llevado a lo largo del camino de los paquetes durante el establecimiento de la reserva y para su transporte se usan los mensajes *PATH* y *RESV* de *RSVP*. Una vez establecida la reserva en el camino de paquetes, el *Número Hash* asignado al flujo se utiliza como *Identificador del Flujo* de paquetes y es transportado en el campo *Flow Label (Etiqueta de Flujo)* de la cabecera de los paquetes *IPv6*. De esta forma, cada vez que un *Encaminador* toma un paquete entrante (ver Figura 3.3), revisa su *Etiqueta de Flujo* y luego utiliza este número como un apuntador a la *Tabla Hash* para encontrar los datos de la *Reserva* y el enlace saliente en la *Tabla de Reservas* (este tipo de clasificación es conocido como *Tagging* –ver sección 2.3.2). Estos datos han sido almacenados previamente durante el proceso de establecimiento de la reserva. El manejo de la *Etiqueta de Flujo* ha sido descrito en [18, 61] y nuestra propuesta está acorde con las políticas allí planteadas.

Debido a que se utiliza un número *Hash* como identificador de un flujo, existe la posibilidad de que haya una *colisión*, es decir, que el número *Hash* que se calculó en el *Host* ya esté asignado en el *Encaminador* a otro flujo. Para solucionar este inconveniente se utiliza una *Tabla de Resolución de Colisiones* (ver Figura 3.3) que contiene la información de las etiquetas que han sido asignadas a varios flujos, así como los datos de la *Quíntupla* de cada uno de estos flujos y un apuntador a la *Tabla de Reservas*. Cuando ocurre una colisión, el *Encaminador* busca en esta tabla cuál de las quíntuplas que tienen la misma *Etiqueta de Flujo* es la que coincide con la del paquete entrante para así encontrar los datos de la reserva y el enlace de salida en la *Tabla de reservas*.

Tabla de Reservas

#Estado Reserva	Enlace saliente	Fuente	Enlace entrante	Salto previo	Recursos Reservados	Tiempo refresco	Estado Solicitud
7	L2	H1	L1	H1	B	1000	Solicitada
8							
9	L2	H4	L6	R2	B	500	Reservada

Tabla hash

Etiqueta de flujo	colisión	#Estado de Reserva
5000	No	7
5025	Si	Nulo

Tabla de resolución de colisiones

Etiqueta de flujo	Quíntupla	#Estado de Reserva
5025	O1,D1,po1,pd1 ,pID1	9
5025	02,D2,po2,pd2, pID2	8

Figura 3.3 Tablas de manejo de las reservas en *IntServ6*

El proceso explicado anteriormente puede observarse en el diagrama de flujo de la Figura 3.4. El *Encaminador* recibe un paquete *IPv6* y analiza su campo *Flow Label*. Si es cero, existe la posibilidad de que sea un mensaje *PATH*, un mensaje *RESV* o un paquete *IPv6* que no pertenece a ningún flujo. Esta decisión la toma en base a la opción *Routing Alert* de *IPv6*, pues los mensajes *PATH* y *RESV* llevan esta opción. Si la *Etiqueta de Flujo* es diferente de cero, se procede a buscar el *valor Hash* de su contenido en la *Tabla Hash* y si se encuentra en ella o en la *Tabla de resolución de colisiones*, en el planificador de paquetes se le dará el tratamiento especificado en la *Tabla de Reservas*. Si no se encuentra en ninguna de las tablas, se le dará tratamiento de un paquete *IPv6* normal.

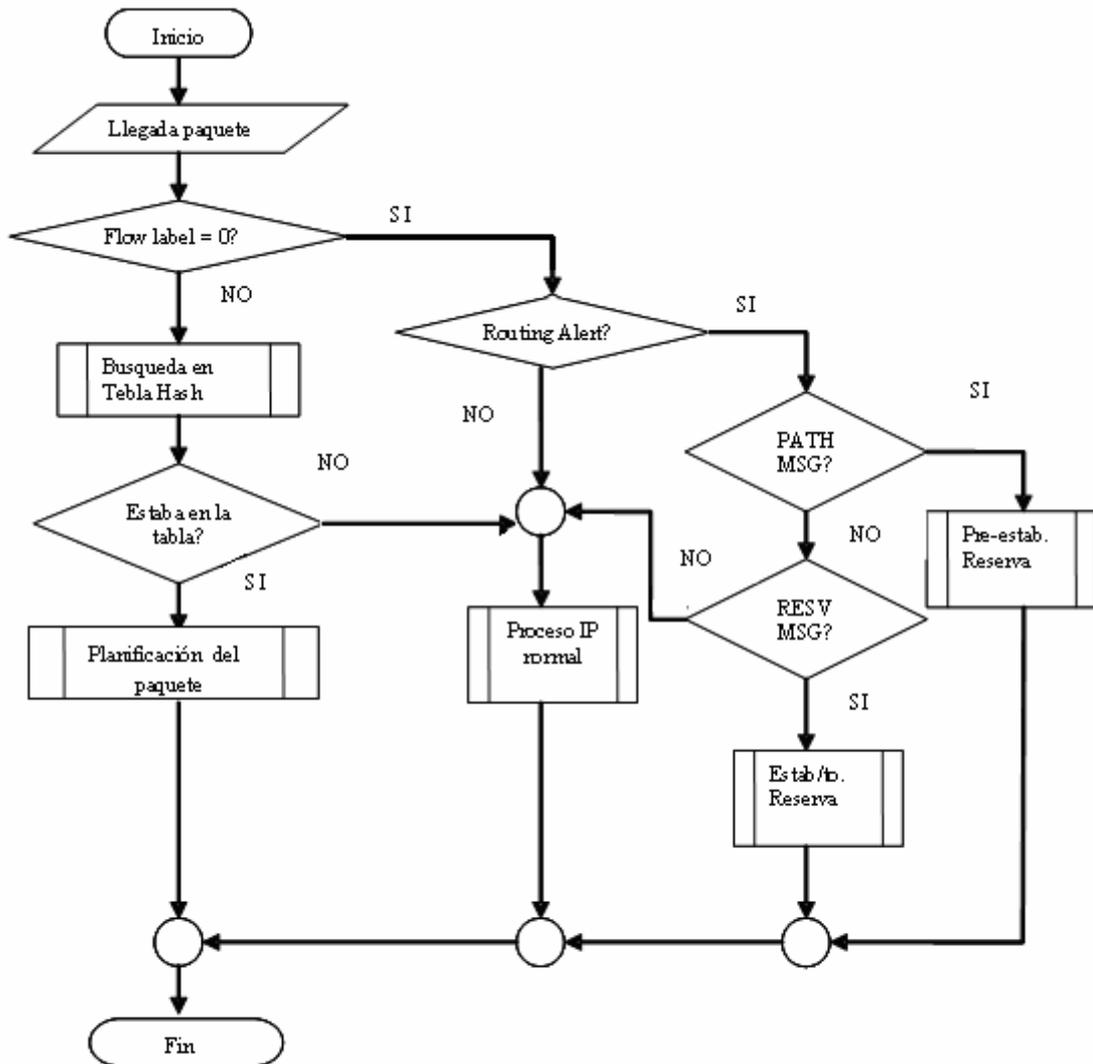


Figura 3.4 Diagrama de flujo del procesamiento de paquetes durante la clasificación en un *Encaminador IntServ6*.

3.3 Reservas dentro de los túneles

Un aspecto importante a resolver con el uso de túneles en redes *IP* es el de mantener reservas de recursos de flujos externos dentro de los túneles. Cuando se utiliza el estándar *IntServ* para el soporte de *Calidad de Servicio*, ocurre un problema debido a la necesidad del cálculo de la *clave Hash* a partir de la *Quíntupla*: dentro del túnel, la *Quíntupla* está escondida dentro del paquete original y no puede ser leída de la cabecera

IP externa. La solución para el estándar *IntServ* se describe en el *RFC 2746* [51]. Esta solución plantea que para diferenciar las reservas dentro de un túnel es necesario transportar los paquetes de las reservas externas dentro de paquetes *UDP* (*User Datagram Protocol*). Por tanto, se asigna un puerto *UDP* para cada reserva dentro del túnel. Como resultado, tal aproximación resuelve un problema de *Capa de Red* en un nivel de *Capa de Transporte*.

Por otro lado, *IntServ6* utiliza una solución simple al problema de la *Quíntupla* escondida dentro de los túneles. Además, *IntServ6* resuelve este problema en la misma *capa de red*. A continuación se detalla la forma en que opera esta propuesta para soportar reservas dentro de túneles.

3.3.1 Reservas dentro de túneles con *IntServ*

El *RFC 2746*, titulado “*Operación de RSVP sobre túneles IP*” [51] describe el método estándar para solucionar el problema del mantenimiento de las reservas dentro de túneles en redes *IP*. En la Figura 3.5 se observa el escenario planteado: Se consideran como *Reservas E2E* (*End-to-End*) o de extremo a extremo, las establecidas en sesiones entre dos *Host* de usuario, mientras que se consideran *Reservas RSVP de Túnel* a las establecidas entre el *Encaminador de Entrada* y el *Encaminador de Salida* del túnel; puede haber una o varias sesiones de túnel entre un mismo par de *Encaminadores* de Entrada y Salida. Además, se consideran como mensajes de señalización *E2E* los mensajes *PATH*, *RESV* y otros enviados entre los *Host* extremos de la sesión *E2E*. Bajo este escenario, el *Encaminador* de entrada y el *Encaminador* de salida del túnel deben estar de acuerdo en las asociaciones existentes entre las *Reservas E2E* y las *Reservas del Túnel* para que los cambios en el estado de la *Reserva E2E* puedan ser mapeados correctamente en cambios en el estado de *Reserva del Túnel*. Por otra parte, tal asociación permite que los errores reportados por los *Encaminadores Intermedios* hacia los extremos del túnel puedan ser transformados en errores reportados por los extremos del túnel a la *Sesión E2E*. Por tanto, el *RFC 2746* describe la forma de crear tal asociación y transportarla de un extremo a otro del túnel.

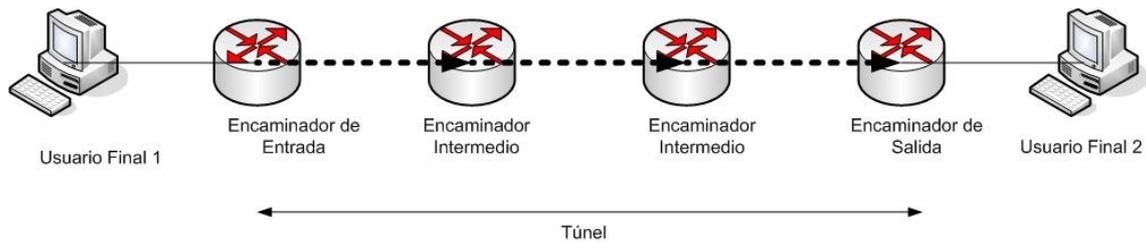


Figura 3.5 Escenario de Reservas dentro de túneles

Así, en este esquema la asociación es creada cuando el *Encaminador de Entrada* del túnel detecta primero un mensaje *RESV* de una sesión *E2E* y entonces fija una nueva *Sesión de Túnel* o la agrega a una *Sesión de Túnel* existente. Esta nueva asociación debe ser llevada al *Encaminador de Salida* del túnel, para que él pueda reservar recursos para las *Sesiones E2E* dentro del túnel. Esto se hace mediante un mensaje *RSVP* conocido como *SESSION_ASOC*, el cual incluye ciertos parámetros de la *Sesión de Túnel* (como el *Puerto Origen* de la sesión), y el identificador de la *Sesión E2E* (*Dirección IP* y *Puerto* en el ordenador de destino) a la que se enlaza la *Sesión del Túnel*.

Finalmente, todo esto se hace para diferenciar las sesiones dentro del túnel, pues todas las *sesiones RSVP* entre los mismos *Encaminadores* de entrada y salida tienen idénticos valores de *Direcciones IP Origen y Destino* y *Número de Puerto UDP Destino*. En consecuencia, en esta solución los paquetes de las *Reservas E2E* se envían dentro del túnel como paquetes *UDP* (lo que se conoce como *encapsulación UDP*) y por tanto, una sesión individual es identificada por el valor del *Puerto Origen*. Así, se aplica la misma *encapsulación UDP* a los mensajes *PATH E2E* y a los mensajes de datos de la sesión. Cuando el *Encaminador de Salida* desencapsula el mensaje *PATH*, deduce la relación entre el puerto *UDP* fuente usado en la encapsulación y la *Sesión RSVP* que está especificada en el mensaje *PATH* original.

3.3.2 Reservas dentro de túneles con IntServ6

Como se mencionó anteriormente, en *IntServ6* las reservas de recursos se diferencian mediante el campo *Etiqueta de Flujo*. Así, cuando se encapsula un paquete en el *Encaminador de entrada* a un túnel (ver Figura 3.6), la etiqueta del paquete *IP* original

permanece dentro de la *cabecera IP interna*. Sin embargo, en *IntServ6* dicho *Encaminador* de entrada al túnel asigna una nueva *Etiqueta de Flujo* en la cabecera externa a cada paquete entrante con reservas dentro del túnel [6, 10]. Esta nueva *Etiqueta de Flujo* se denomina *Etiqueta de Túnel* y se utiliza para diferenciar las reservas dentro del túnel. Para la asignación de las etiquetas en los túneles se deberían tener en cuenta los campos de la *Quíntupla*. Sin embargo, los *puertos fuente y destino* no son utilizados en los *Encaminadores*, por lo que para asignar la *clave hash* de la *etiqueta del túnel* se podrían contemplar varias alternativas:

- a. *Utilizar sólo las direcciones IP de origen y destino*: No es una buena solución pues esta opción arrojaría siempre la misma *clave hash* ya que las direcciones *IP* de origen y destino son siempre las mismas entre el mismo par de *Encaminadores*.
- b. *Utilizar los mismos puertos de la reserva E2E*: Esta solución arrojaría muchas menos *claves hash*, pues los puertos son de 16 bits de longitud. Además se requiere la lectura de campos de la *capa de transporte*.
- c. *Utilizar números de puertos aleatorios*: Al igual que la opción anterior, esta solución arrojaría menos *claves hash*, aunque dichas claves estarían distribuidas más uniformemente. También requiera la lectura de campos de la *capa de transporte*.
- d. *Asignar una etiqueta de túnel aleatoria y uniformemente distribuida sin tener en cuenta la Quíntupla*: Esta solución es la más adecuada pues permite asignar una *Clave hash* que se distribuye uniformemente y además no requiere la lectura de campos de la capa de transporte.

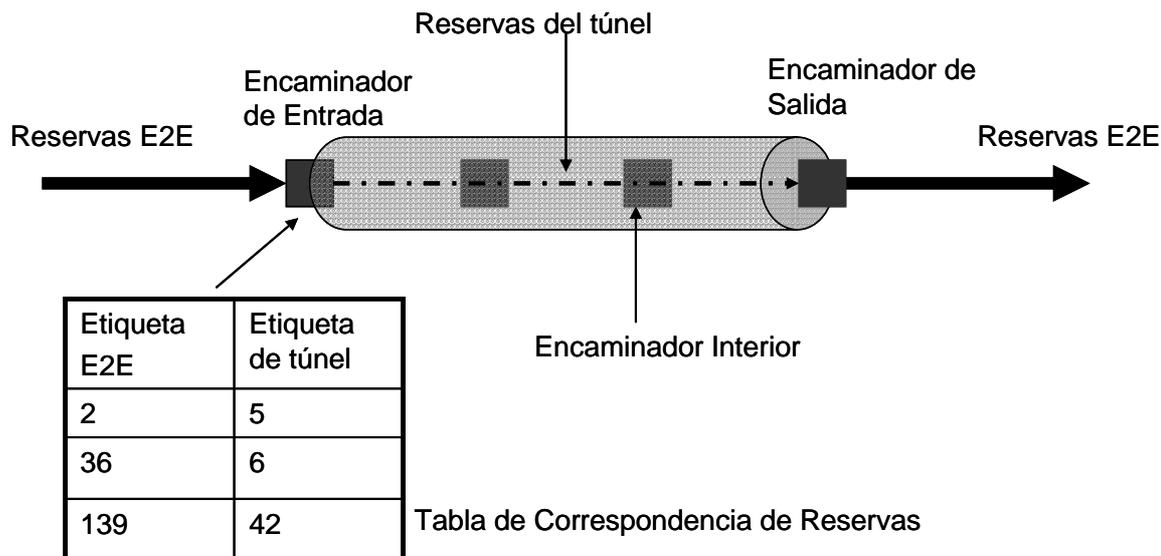


Figura 3.6 Diferenciación de Reservas dentro de túneles con *IntServ6*

Adicionalmente, debe establecerse la reserva de recursos dentro del túnel antes de iniciar la transmisión de paquetes de datos de la misma. Dicha reserva de recursos debe ser suficiente para mantener las reservas de todos los flujos entrantes al túnel. El método para calcular el número requerido de recursos dentro del túnel no se estudia en este documento.

Para obtener un mejor entendimiento del proceso, primero se definen dos tipos de reservas (ver Figura 3.6): Las *Reservas Extremo a Extremo (E2E, End-to-End)*, que son las reservas asignadas antes y después del túnel, y las *reservas del túnel*, que son las reservas de recursos asignadas en cada uno de los *Encaminadores* del túnel (*Encaminadores de Entrada y Salida* y *Encaminadores Interiores*).

Inicialmente, cuando se establece una reserva dentro de un túnel, se deberán intercambiar mensajes *PATH* y *RESV* entre el *Encaminador de Entrada* al túnel y el *Encaminador de Salida* del túnel. El *Encaminador de Entrada* del túnel especifica los recursos requeridos por medio del objeto *SenderTSpec* del protocolo *RSVP*, que es enviado en el mensaje *PATH* (ver sección 2.3.1). Por su parte, el *Encaminador de Salida* del túnel debe decidir exactamente la cantidad de recursos que requiere la reserva. Esta información es enviada hacia el *Encaminador de Entrada* del túnel mediante un mensaje *RESV*. Cuando se establece la reserva dentro del túnel, debe

agregarse una nueva entrada a la *Tabla de Correspondencia de Reservas* del *Encaminador de Entrada* al túnel y en la *Tabla Hash* de cada *Encaminador Interior* del túnel. Cada entrada en la *Tabla de Correspondencia de Reservas* tiene la *Etiqueta E2E* y la *Etiqueta de Túnel* correspondiente. En este momento, el camino del túnel está listo para el envío de paquetes de datos.

Por otro lado, los mensajes *PATH* y *RESV* para las reservas *E2E* viajan dentro del túnel como mensajes encapsulados. Por tanto, estos mensajes no son reconocidos por los *Encaminadores Interiores* del túnel, mientras que los mensajes *PATH* y *RESV* de la *Reserva Interna del Túnel* viajan sin encapsular y por tanto, son identificados por los *Encaminadores Interiores del Túnel*.

3.4 Agregación de flujos

La *Agregación de Flujos* en redes *IP* permite una organización jerárquica de una *Internet*. Cuando se utiliza *Agregación de Flujos*, los *Encaminadores* conmutan paquetes utilizando la dirección del *Dominio de Destino* en lugar de la dirección *IP* del odenador del usuario final. Esta característica agiliza el proceso de conmutación de paquetes en dichos *Encaminadores* a la vez que permite reducir el tamaño de las tablas de Encaminamiento. Esto conlleva a un mejoramiento de la escalabilidad de la red. Por tanto, una solución que soporte *QoS* y que se ajuste fácilmente a la característica de *Agregación de Flujos* de *Internet*, tendrá una mayor escalabilidad.

La solución para esta situación en el estándar *IntServ* se encuentra descrita en el *RFC* 3175 [69]. Esta aproximación utiliza redes de transporte *DiffServ* para permitir la *Agregación de Flujos*. Así, utiliza un *Encaminador Frontera* de *Agregación* y un *Encaminador Frontera* para *Desagregación*. Estos permiten establecer la *Correspondencia de Reservas IntServ E2E* dentro de las clases de *Servicios Diferenciados* de la *Red de Transporte*. Por tanto, el mismo estándar *IntServ* no tiene esta capacidad y debe usar otra tecnología para solucionar este problema. El problema para soportar la *Agregación de Flujos* en *IntServ* es debido a que un *Encaminador* debe realizar el cálculo de la *Clave Hash* a partir de la *Quíntupla* para cada paquete entrante. Por tanto, se requiere usar todos los bits de dirección *IP* destino para soportar la reserva

de recursos. Por otro lado, *IntServ6* permite la *Agregación de Flujos* mediante un método similar al *Apilado de Etiquetas* en *MPLS* (*MPLS Label Stacking*). A continuación se describirá la solución estándar actual y luego se describirá la solución de *IntServ6* para la *Agregación de Flujos*.

3.4.1 Agregación de flujos con el estándar actual

La solución que se propone en el *RFC 3175* [69] (ver Figura 3.7) involucra la *Agregación* de diferentes reservas que cruzan una *Región de Agregación* y comparten *Encaminadores* comunes de ingreso y egreso con una reserva grande entre ellos. Se define una *Región de Agregación* como un conjunto continuo de sistemas capaz de realizar *Agregación RSVP* a lo largo de cualquier posible ruta a través del mismo.

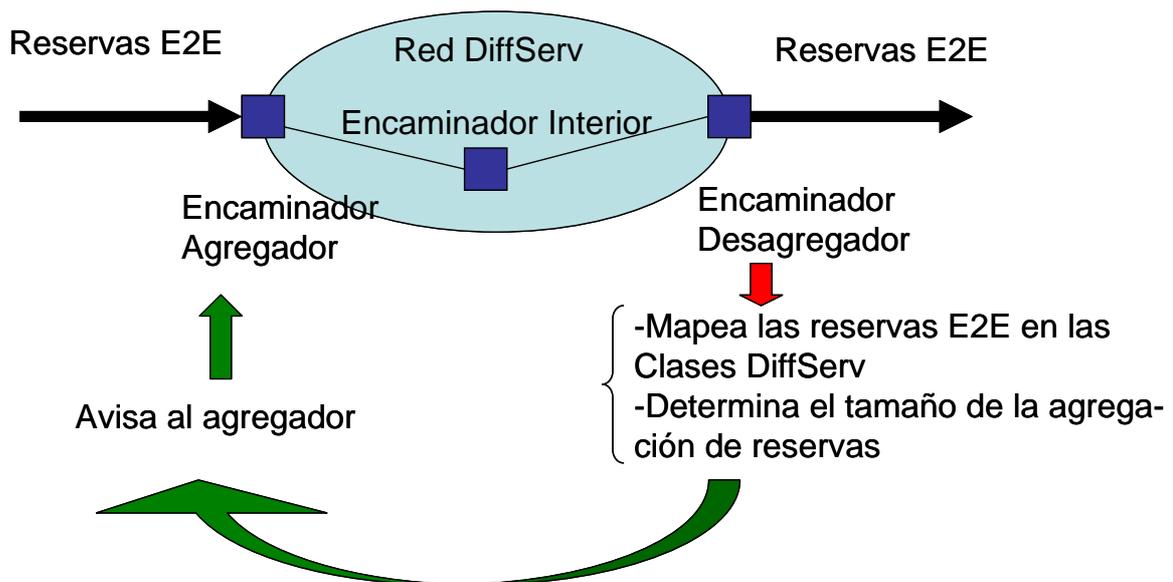


Figura 3.7 Agregación de flujos RSVP estándar

Las interfaces de comunicación caen dentro de dos categorías con respecto a una *Región de Agregación*; ellas son *Exteriores* a una *Región de Agregación*, o *Interiores* a una *Región de Agregación*. Los *Encaminadores* que tienen al menos una interfaz dentro de la región caen en una de tres categorías con respecto a una sesión RSVP dada; ellos *Agregan*, *Desagregan* o están entre un *Agregador* y un *Desagregador*.

La *Agregación* depende de la capacidad de esconder los mensajes *RSVP E2E* de los *Encaminadores* con capacidad *RSVP* dentro de la *Región de Agregación*. Para obtener esto, el número de protocolo *IP* en los mensajes de la *Reserva E2E* (*Path*, *PathTear*, y *ResvConf*), se cambia del valor *RSVP* (46) a *RSVP-E2E-IGNORE* (134) una vez entran en la *Región de Agregación*, y se restauran a *RSVP* en el punto *Desagregador*. Estos mensajes son ignorados por cada *Encaminador* dentro de la *Región de Agregación* cuando son enviados por una *Interfaz Interior*. Ya que el *Encaminador* que desagrega percibe que el salto previo en tales mensajes es el *Encaminador* que agrega, el mensaje *RESV* y otros no requieren esta modificación; estos son *Unicast* desde un salto *RSVP* a otro de todas maneras.

Los recursos solicitados de cada una de las *Reservas E2E* son sumados y el total es transportado en los elementos de información correspondientes (*SenderTSpec* y *FlowSpec*) en los mensajes *PATH* y *RESV* del agregado. Los mensajes *PATH* de los *Agregados* son enviados desde el *Agregador* hasta el *Desagregador* usando el número de protocolo *IP* normal de *RSVP*. Los mensajes *RESV* de los *Agregados* son enviados hacia atrás desde el *Desagregador* hacia el *Agregador*, estableciendo así una *reserva de agregado* para el conjunto de flujos *E2E* que usan este par *Agregador-Desagregador*.

Debido al uso de mecanismos de *Servicios Diferenciados* para la clasificación y planificación del tráfico transportado por las *Reservas de Agregados*, se reduce la cantidad de recursos para la clasificación y la reserva. Esta cantidad no es sólo independiente del número de *Reservas E2E*, también es independiente del número de *Reservas de Agregados* en la *Región de Agregación*. Se usan uno o más *DSCPs* (*Differentiated Services Code Point*) para identificar el tráfico cubierto por las *Reservas de Agregados* y se usan uno o más *PHBs* de *DiffServ* para ofrecer el *Tratamiento de Retransmisión* requerido para este tráfico. Podría haber más de una *Reserva de Agregado* entre el mismo par de *Encaminadores*, cada una representando diferentes clases de tráfico y cada una usando un *DSCP* diferente y un *PHB* diferente.

3.4.2 Agregación de flujos con IntServ6

El procedimiento para gestionar la *Agregación de Flujos* en *IntServ6* [6] puede observarse en la Figura 3.8. El principio básico es el uso del *Entunelamiento IP*. Cuando un alto número de flujos viajan desde un mismo *Dominio Fuente* a un mismo *Dominio Destino*, puede utilizarse una red de transporte basada en *IntServ6* para llevar el tráfico agregado. Para hacer esto, es necesario que un *Encaminador Frontera de Entrada* en la red de transporte *IntServ6* coloque todos los paquetes de un mismo agregado en un túnel entre él mismo y el *Encaminador Frontera de Salida*. Así, todos los flujos que pertenecen a un mismo *Agregado* tendrán la misma *Etiqueta de Flujo* en la cabecera externa *IPv6* del túnel. Como resultado, cada *Encaminador Frontera de Entrada* en la red de transporte *IntServ6* debe tener una *Tabla de Correspondencia de Etiquetas*, la cual almacena las *Etiquetas de Flujo E2E* y su correspondiente *Etiqueta de Flujo del Agregado*. Por tanto, el *Encaminador Frontera de Salida* sólo deberá extraer los paquetes *IP E2E* del túnel y transmitirlos.

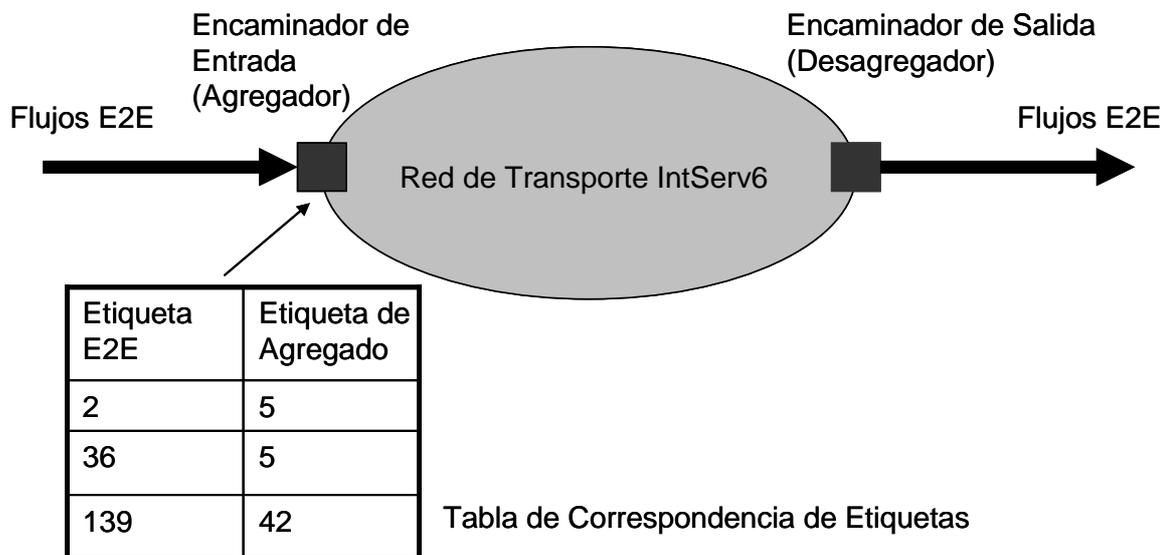


Figura 3.8 Agregación de flujos en *IntServ6*

En la Figura 3.8 se observa que un agregado es un conjunto de flujos externos que utilizan la misma *Etiqueta de Túnel*. Es decir, se establece la misma reserva dentro del túnel para uno o más flujos. Adicionalmente, dado que pueden mapearse varios *Flujos E2E* a una misma *Etiqueta de Túnel*, el *Encaminador de Salida del Túnel* deberá calcular el número de recursos requeridos para mantener la misma *QoS* para todos los

flujos E2E dentro del *Agregado*. El método para calcular estos recursos no hace parte del alcance de este documento.

Adicionalmente, podría ocurrir que aparezca un nuevo flujo o desaparezca un flujo existente dentro del agregado. Para solucionar esta situación, *IntServ6* introduce dos nuevos mensajes de señalización *RSVP*. Cuando aparece un nuevo flujo, el *Encaminador Frontera de Entrada* enviará al *Encaminador Frontera de Salida* un mensaje *PATH* con un objeto *RESIZE*. Este mensaje transporta la *Etiqueta del Agregado de Flujos*, los recursos adicionales requeridos y un bit *SIGN* en estado *ALTO* que indica a los *Encaminadores* del camino que los recursos solicitados deben ser adicionados a la reserva del agregado especificado en caso de ser aceptada la solicitud.

Una vez que el *Encaminador de Salida* del túnel determina si estos recursos pueden ser adicionados, deberá confirmar mediante un mensaje *RESV* que transporta el mismo objeto *RESIZE*. Este mensaje atraviesa el camino del túnel hasta llegar al *Encaminador de Entrada* del túnel. Cuando la adición de recursos ha sido confirmada, el *Encaminador de Entrada* al túnel asigna una nueva entrada en la *Tabla de Correspondencia de Reservas* y además modifica las otras tablas mencionadas anteriormente.

Cuando desaparece un flujo existente el *Encaminador de Entrada* del túnel, o el *Encaminador de Salida* del túnel, enviará un mensaje *PATH* o *RESV* según sea el caso. Este mensaje transporta el objeto *RESIZE* especificando la cantidad de recursos que deben deducirse de la reserva de recursos del agregado y el valor de la *Etiqueta de Flujo* que debe eliminarse de la *Tabla de Correspondencia de Reservas* en el *Encaminador de Entrada* del túnel. Adicionalmente, el bit *SIGN* debe estar en el Estado *BAJO* para informar que estos recursos deberían eliminarse de las tablas.

3.5 Interconexión con redes de transporte MPLS

En la actualidad un aspecto muy importante de las redes *IP* es la interconexión con redes *MPLS*. Por tanto, cualquier arquitectura de *QoS* debería proveer soluciones que permitan este tipo de interconexión. En el caso de *IntServ*, este aspecto no había sido

estandarizado, pero recientemente salió un nuevo estándar conocido como *RFC 4804* [70], que describe la forma en que deben interactuar los *LER (Label Edge Router)* de una red *MPLS* con redes clientes basadas en *IntServ*. Este permite de alguna manera automatizar el proceso de interconexión entre las dos tecnologías cuando se utiliza el protocolo de distribución de etiquetas *RSVP-TE* en la red *MPLS*. Sin embargo, esta solución no especifica cómo se establecen nuevas reservas y su correspondencia entre los túneles *RSVP-TE* en la red *MPLS* y las reservas de la red *IntServ*. Tampoco se menciona cómo solucionar el problema del cálculo de la *clave hash* a partir de la *Quíntupla* en el *LER* de entrada a la red *MPLS* para identificar los flujos entrantes y asignarles una etiqueta de entrada a la red *MPLS* o el cálculo de la *clave hash* para retransmitir los paquetes hacia la red *MPLS*, teniendo que resolver colisiones si estas ocurriesen.

Por su parte, *IntServ6* permite una fácil interconexión con redes *MPLS* [6]. Debido que las redes *MPLS* proveen recursos a los usuarios y dichos recursos deben cumplir con el Acuerdo Pre-Definido de Nivel de Servicio (*SLA- Service Level Agreement*), no es necesaria ninguna negociación en línea entre la red cliente y las redes de transporte *MPLS*. Por tanto, una red *IntServ6* cliente sólo debe enviar sus paquetes a la red de transporte *MPLS* y el *LER* de la red *MPLS* realizará los mismos procedimientos estándar que deben ejecutarse con cualquier otra red *IP*. Es decir, el *LER* de la red *MPLS* debe leer la cabecera de los paquetes *IPv6* y enviar los paquetes de acuerdo con el tratamiento pre-definido en el *SLA*.

Adicionalmente, pueden utilizarse las características de *IntServ6* para mejorar la interconexión con las redes de transporte *MPLS*. Si la red *MPLS* está basada en *Servicios Integrados* para el soporte de *QoS* y utiliza el protocolo *RSVP-TE* para realizar la distribución de etiquetas, se tendrá un ambiente similar al de la Figura 3.9.

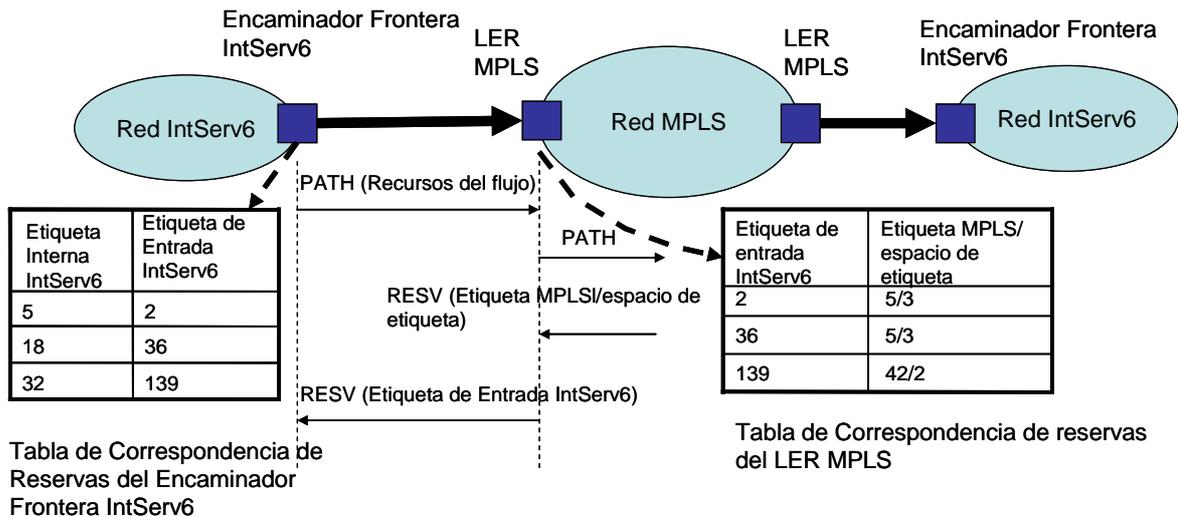


Figura 3.9 Interconexión de una red *IntServ6* a una red *MPLS* utilizando Correspondencia de reservas automática.

Bajo este entorno, las reservas *IntServ6* pueden mapearse en la red *MPLS* mediante dos *Tablas de Correspondencia de Reservas* ubicadas dentro del *LER MPLS* y el *Encaminador Frontera* de la red *IntServ6* respectivamente. Por su parte, la tabla ubicada en el *LER MPLS* almacena tanto las etiquetas de entrada de la red *IntServ6* como las etiquetas de entrada *MPLS* utilizadas dentro de la red *MPLS*. De esta manera, cuando llegan paquetes desde la red *IntServ6*, el *LER MPLS* lee el campo de *Etiqueta de Flujo* y busca esta etiqueta en la *Tabla de Correspondencia de Reservas* para obtener el espacio *MPLS/Label* asignado a ella. Para agregar nuevas reservas a las *Tablas de Correspondencia de Reservas*, el *Encaminador Frontera IntServ6* debe solicitar una *etiqueta de entrada IntServ6* al *LER MPLS*. Este procedimiento se realiza enviando un mensaje *PATH* que contiene los recursos requeridos para el nuevo flujo. El *LER MPLS* realiza la asignación de etiquetas interna utilizando los procedimientos del protocolo *RSVP-TE*. Entonces, el *LER MPLS* informa mediante un mensaje *RESV* al *Encaminador Frontera IntServ6* cuál número de *Etiqueta de Flujo* debe ser usado para este flujo. En consecuencia, el *Encaminador Frontera IntServ6* agrega una nueva entrada dentro de la *Tabla de Correspondencia de Reservas*. Dicha tabla almacena la *Etiqueta de Flujo* usada internamente en la red *IntServ6* y la *Etiqueta de Flujo IntServ6 de entrada* que fue asignada por el *LER MPLS*. Este procedimiento fue diseñado para evitar que los *LER's MPLS* tengan que resolver las colisiones de las claves *Hash*. De

esta manera, esta función será realizada únicamente dentro de los *Encaminadores Frontera IntServ6*.

Esta solución agiliza el proceso de envío de paquetes en el *LER MPLS* ya que, a diferencia del procedimiento estándar definido en el *RFC 4804*, no debe leerse la cabecera *IP* completa. En su lugar, el *LER MPLS* sólo debe leer el campo de *Etiqueta de Flujo* y luego enviar los paquetes hacia la red *MPLS* usando la etiqueta de entrada correspondiente. Utilizando esta configuración, las redes *MPLS* pueden adaptar sus recursos internos a los requisitos de recursos de la red cliente en una forma automática que cambia fácilmente las reservas de recursos en el tiempo.

3.6 Extensiones para el soporte a la movilidad

Las redes móviles de próxima generación estarán basadas en el protocolo *IPv6*. Estas redes deberán soportar servicios con *Tráfico Elástico* tales como *Transferencia de Archivos* y servicios de tiempo real con información como voz y video, por lo que una solución de *QoS* para este tipo de redes debe cumplir una serie de requisitos que fueron descritos antes en la sección 2.6. En consecuencia, *IntServ6* tiene extensiones para estos entornos móviles (a esta variante de *IntServ6* que comprende las extensiones de movilidad se le denomina *IntServ6 Móvil -en Inglés, Mobile IntServ6*). Tales extensiones fueron introducidas recientemente en [10]. El principal problema a resolver sobre las redes móviles basadas en *IntServ6* es que la dirección *CoA* (*Care of Address*) cambia cuando ocurre un *Traspaso*. Como resultado, la *Quíntupla* cambia y por tanto, el número transportado en la *Etiqueta de Flujo* y utilizado para identificar la reserva cambia también. Como consecuencia, la reserva de recursos fallará. Para evitar este problema, *IntServ6* presenta unas extensiones para ambientes móviles que permiten establecer una nueva reserva sobre el nuevo camino de paquetes y luego, deshacer la reserva del camino anterior. Algunas soluciones planteadas anteriormente a este problema ya fueron discutidas en la sección 2.6.

Otro problema importante que aparece es que el tiempo de duración (*Latencia*) del *Traspaso* se incrementa debido a los tiempos adicionales necesarios para establecer la nueva reserva y eliminar la reserva anterior. Por tanto, es deseable utilizar tecnologías

que permitan reducir el *Tiempo de Traspaso*. Una alternativa es utilizar el protocolo *Fast Handoff* [71] para complementar al protocolo de movilidad *MIPv6*. Esta alternativa presenta un buen rendimiento para aplicaciones multimedia, tal como se ha comprobado en estudios experimentales como el presentado en [72]. Sin embargo, además de utilizar el protocolo *Fast Handoff*, se puede reducir aún más el tiempo de *Latencia del Traspaso* si se utiliza un protocolo de micro-movilidad como el *HMIPv6* [72]. Por tanto, *IntServ6* debería operar sobre una *Red Jerárquica* basada en el protocolo de movilidad *HMIPv6* [3] y se podría complementar con el protocolo *Fast Handoff* para reducir aún más el *Tiempo de Traspaso*. Con el uso de *HMIPv6*, el segmento del camino que cambia cuando se realiza el *Traspaso* se reducirá a unos pocos *Encaminadores* ubicados muy cerca del *Nodo Móvil*. De esta manera, el cambio en el camino de la reserva se reduce a una pequeña parte del camino de los paquetes.

Un aspecto clave a tener en cuenta cuando se tiene una solución que soporta *QoS* en una red con soporte de movilidad con protocolos como *MIP* y *HMIP* es el *Acoplamiento* de las dos tecnologías. Por *Acoplamiento* se entiende la implementación de un mecanismo de señalización que permite a un protocolo informar a otro protocolo que ha ocurrido un evento especial [73]. En el caso de la operación de *IntServ6* en redes móviles, se debe elegir la forma en que el protocolo de movilidad (*MIP*) debe informar al mecanismo de *QoS* (*IntServ6*) sobre un cambio en la ubicación del móvil debido a un *Traspaso*. Existen tres formas de acoplamiento [73]:

- *No Coupling*: Los protocolos trabajan independientemente y sólo se comunican eventos externos (tales como cambios en el camino de los paquetes).
- *Loose Coupling*: Si ocurre un evento bajo un protocolo, este deberá producir alguna acción sobre el otro.
- *Hard Coupling*: La información relacionada con ambos protocolos (*QoS* y movilidad) es transportada como un todo, ya sea como una extensión de un protocolo o como un protocolo que gestiona *QoS* y movilidad.

Para realizar una elección acerca de la opción más adecuada para el acoplamiento, se deben tener en cuenta tres criterios: eficiencia, complejidad y aplicabilidad. Con la opción “*No coupling*”, se garantizan la transparencia y la independencia de los mecanismos. Sin embargo, no hay mejoras en el rendimiento y en la eficiencia del

sistema. Por otro lado, la opción “*Hard Coupling*” mejora la eficiencia del sistema pero a cambio se disminuye la independencia, lo que atenta contra la filosofía de diseño por capas en los sistemas de comunicaciones. Finalmente, la opción “*Loose Coupling*” es una opción intermedia.

Como resultado de estas condiciones, la arquitectura de red propuesta (ver Figura 3.10) utiliza el protocolo *HMIPv6* para el soporte de la movilidad. Otra característica importante de *IntServ6 Móvil* es que utiliza acoplamiento de tipo “*Loose Coupling*”, por lo que hace modificaciones mínimas a los protocolos de movilidad estandarizados. Por tanto, para soportar la movilidad de las reservas, *IntServ6 Móvil* utiliza una solución similar a la propuesta por Paskalis [52] (ver sección 2.6.2) pero con algunas modificaciones. Así, *IntServ6 Móvil* hace uso de *Agentes de Movilidad HMIPv6*, los cuales soportan el protocolo *HMIPv6*, y además utiliza *Agentes de Movilidad de las Reservas (RMA-Reservations Mobility Agents ó Agentes MISv6)* ubicados en los *MAP (Mobility Anchor Point)* de la red *HMIPv6*. Cuando ocurre un evento de movilidad, es decir, hay un *Traspaso*, los *Agentes de Movilidad HMIPv6* alertan a los *RMAs* y estos inician un procedimiento de *Traspaso* de la respectiva reserva.

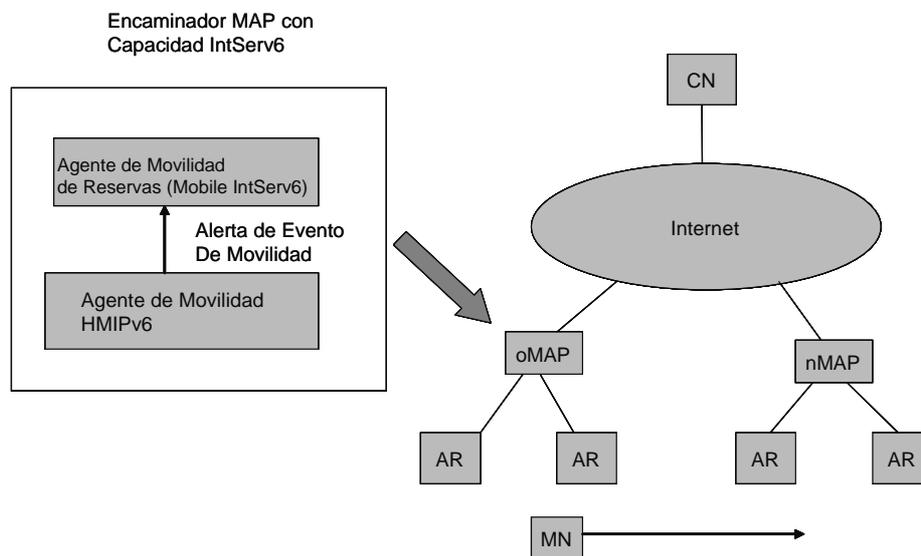


Figura 3.10 Arquitectura de la red *IntServ6* Móvil.

Bajo el entorno de movilidad planteado, podrían surgir diferentes tipos de *Traspaso*. Un primer tipo de *Traspaso* sería un *Traspaso Local*; en esta situación sólo habrá

cambios de *Encaminadores de Acceso* (AR, *Access Routers*) pertenecientes a un mismo MAP, por lo que no hay cambio de MAP. Un segundo tipo de *Traspaso* es el *Traspaso Inter-Dominio*. En este caso, el *Nodo Móvil* cambia de su antiguo AR (*oAR*, *old Access Router*) a un nuevo AR (*nAR*, *New Access Router*) perteneciente a otro MAP (*nMAP*, *new Mobility Anchor Point*). Las principales situaciones se describen en las figuras 3.11 y 3.12, donde se muestran un *Traspaso local* y un *Traspaso Inter-Dominio* (con el fin de simplificar esta descripción, no se tiene en cuenta el uso del protocolo *Fast Handoff*). Como se observa en la Figura 3.11, cuando es exitoso un *Traspaso Local*, el *Agente de Movilidad MAP* recibe un *mensaje de actualización de la posición* (*BU-Binding Update*) con la *nueva LCoA* (*nLCoA*, *new Local Care of Address*) desde el *Nodo Móvil*. En este momento, el *Agente de movilidad HMIPv6* envía un mensaje *BACK* (*Binding Acknowledge*) al *Nodo Móvil* y también alerta al *RMA* sobre un evento de movilidad. En consecuencia, el *RMA* inicia el establecimiento de una nueva reserva y envía un mensaje *PATH* hacia el *Nodo Móvil*. Una vez que la nueva reserva se ha establecido, la reserva anterior es eliminada mediante el envío de mensajes *PATH Tear* y *RESV Tear* entre el *RMA* y el *AR* antiguo (*oAR*). Por simplicidad, en la figura no se detalla el establecimiento de la reserva dentro del túnel entre el *MAP* y el *nAR* y se asume que se realiza implícitamente cuando se establece dicho túnel. El proceso de establecimiento de la reserva dentro del túnel sigue las reglas descritas en la sección 3.3.2.

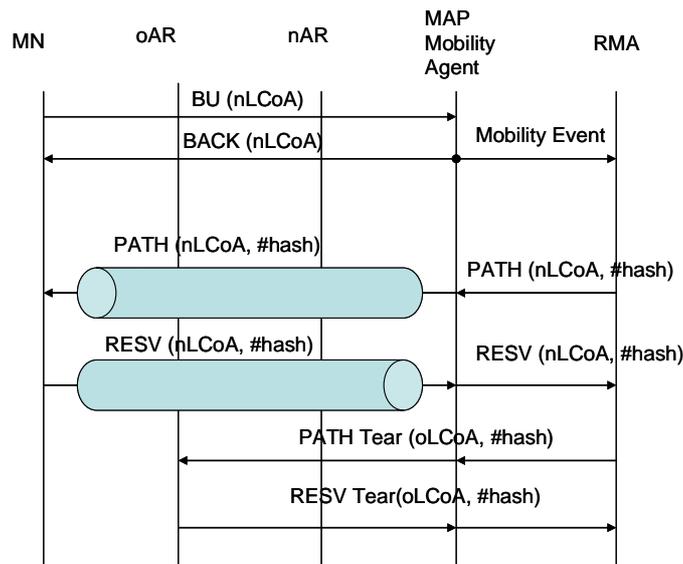


Figura 3.11 *Traspaso Local* en una red *IntServ6 Móvil*.

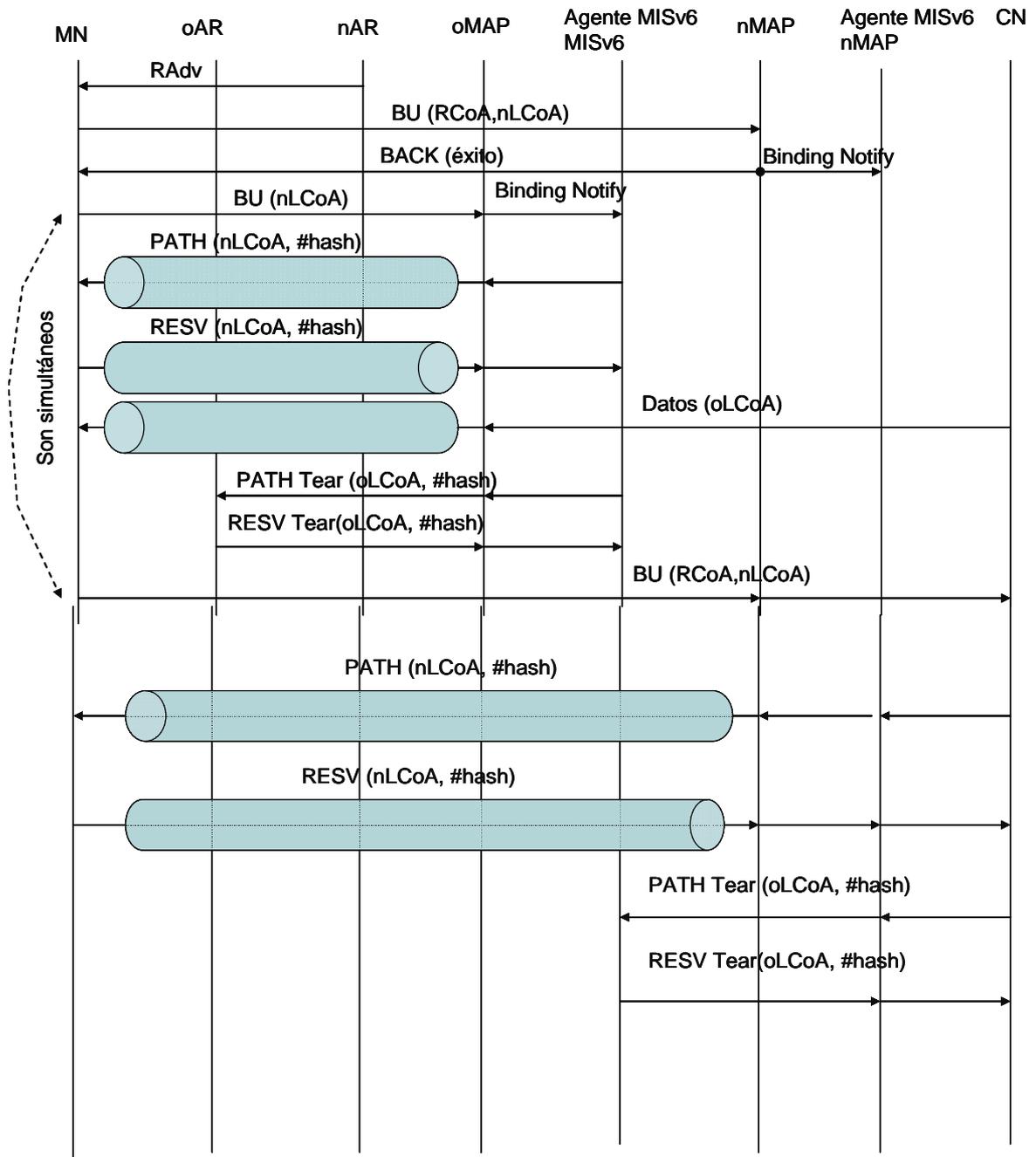


Figura 3.12 Traspaso Inter-Dominio en una red IntServ Móvil.

3.7 Cumplimiento de las especificaciones de la Etiqueta de Flujo en la propuesta IntServ6

En la Tabla 3.1 se describen los requisitos de la especificación de etiquetas de flujo que se encuentran consignadas en el RFC 3697 [61] y la forma en que se cumplen en la propuesta IntServ6. Las especificaciones que tienen que ver con seguridad no son tratadas en la forma actual de esta propuesta y son parte de las líneas futuras para la mejora de la misma.

Tabla 3.1 Cumplimiento de las especificaciones de las *Etiquetas de Flujo IPv6*

Requerimiento del RFC 3697	Cumplimiento en IntServ6
Un flujo podría estar compuesto de todos los paquetes en una conexión de transporte específica o un flujo de paquetes de una aplicación multimedia. Sin embargo, un flujo no necesariamente tiene una relación 1:1 con una conexión de transporte.	En nuestra propuesta un flujo tiene una relación 1:1 con las conexiones de la capa de transporte debido a que se utiliza la Quintupla para asignar el valor de la <i>Etiqueta de Flujo</i> .
Tradicionalmente, los clasificadores de flujos se han basado en la <i>Quíntupla</i> del origen y el destino, compuesta por las <i>direcciones fuente y destino</i> , los <i>puertos fuente y destino</i> y el <i>tipo de protocolo</i> . Sin embargo, algunos de estos campos no están disponibles debido a la fragmentación o a la encriptación. Además, el buscarlos en una cadena de <i>cabeceras de opciones de IPv6</i> podría ser ineficiente. Adicionalmente, si los clasificadores dependen sólo de las cabeceras de la <i>capa IP</i> , la introducción posterior de los protocolos de cabecera de transporte alternativos sería más fácil. El uso de la <i>3-tupla</i> de la <i>Etiqueta de Flujo</i> y los campos de <i>dirección fuente y destino</i> permite la clasificación eficiente de flujos IPv6, donde sólo se usan campos en posiciones fijas de la cabecera principal de IPv6.	Para la clasificación se utiliza la <i>Etiqueta de Flujo</i> solamente (no se usa la <i>3-tupla</i>). Esto se debe a que el valor de la <i>Etiqueta de Flujo</i> es calculado a partir de la <i>Quíntupla</i> . En IntServ6 no se requiere que en los <i>Encaminadores</i> se revisen las cabeceras de transporte pues la <i>Etiqueta de Flujo</i> es calculada y asignada por el <i>Host Fuente</i> mientras que los <i>Encaminadores</i> sólo revisan la <i>Etiqueta de Flujo</i> de la cabecera IPv6.
El campo de <i>Etiqueta de Flujo</i> en la cabecera IPv6 es usado por una fuente para etiquetar paquetes de un flujo. Para indicar que los paquetes no hacen parte de ningún flujo se utiliza la <i>Etiqueta de Flujo Cero</i> . Los clasificadores de paquetes usan la <i>tripleta</i> de los campos <i>Etiqueta de Flujo</i> , <i>Dirección Fuente</i> y <i>Dirección Destino</i> para identificar a qué flujo en particular pertenece un paquete.	La <i>Etiqueta de Flujos</i> se asigna en el Host origen. Cuando un paquete no pertenece a ningún flujo, su valor de <i>Etiqueta de Flujo</i> es cero.
Los paquetes son procesados por los nodos (que han sido fijados previamente con el estado específico del flujo) en la forma específica de su flujo. La naturaleza del tratamiento y los métodos para el establecimiento del estado del flujo están fuera del alcance de este documento.	La clasificación se hace solamente con base en la <i>Etiqueta de Flujo</i> (no se usan las direcciones Origen y Destino), pues esta se ha obtenido a partir de la <i>Quíntupla</i> y ya lleva esta información. Los paquetes pertenecientes a un flujo son procesados de acuerdo con la reserva de recursos establecida previamente.
El valor de la <i>Etiqueta de Flujo</i> fijado por la fuente debe ser entregado sin alteración al nodo o nodos destino.	El valor de la <i>Etiqueta de Flujo</i> no se altera en el camino de los paquetes.

Requerimiento del RFC 3697	Cumplimiento en IntServ6
<p>Los nodos IPv6 no deben asumir ninguna propiedad matemática de los valores de <i>Etiqueta de Flujo</i> asignados por los <i>nodos fuente</i>. El desempeño de los <i>Encaminadores</i> no debe depender de la distribución de los valores de la <i>Etiqueta de Flujo</i>. Especialmente, los bits de la <i>Etiqueta de Flujo</i> solos son material muy pobre para una clave hash.</p>	<p>Esto significa que los <i>Encaminadores</i> no deben tomar decisiones de tratamiento de los paquetes a partir de cálculos matemáticos con la <i>Etiqueta de Flujo</i>. Esto se cumple en <i>IntServ6</i>.</p>
<p>El uso del campo de <i>Etiqueta de Flujo</i> no necesariamente señala ningún requisito en el ordenamiento de los paquetes. Especialmente, la etiqueta cero no implica que se acepte un ordenamiento.</p>	<p>La <i>Etiqueta de Flujo</i> por sí misma no es un buen material para calcular una <i>clave hash</i> a partir de ella, pero sí constituye un buen medio de transporte para dicha clave. El valor de la <i>Etiqueta de Flujo</i> es asignado mediante una <i>función hash</i> a partir de la <i>Quíntupla</i>. Esta <i>función hash</i> se supone debe asignar valores uniformemente en el rango de <i>claves hash</i>, por lo que esto no afecta que ciertos valores tengan un procesamiento más rápido.</p>
<p>Si un nodo IPv6 no provee tratamiento específico por flujos, debe ignorar el campo cuando recibe o envía un paquete.</p>	<p>La búsqueda en la <i>tabla hash</i> se hace usando <i>Tagging</i>, por lo que sólo emplea una operación de lectura independiente del valor de la <i>Etiqueta de Flujo</i>. Por otro lado, la ubicación de los flujos en la <i>tabla de resolución de colisiones</i> es totalmente aleatoria.</p>
<p>El <i>nodo fuente</i> debe proveer medios para que las aplicaciones y los protocolos de transporte puedan especificar los valores de las etiquetas de flujo y definir qué paquetes constituyan un flujo.</p>	<p>En <i>IntServ6</i>, el orden de asignación de los valores de <i>Etiqueta de Flujo</i> es <i>pseudo-aleatorio</i> mediante una <i>clave hash</i> y no se hace ningún tratamiento especial debido al orden que lleve la <i>Etiqueta de Flujo</i>.</p>
<p>El <i>nodo fuente</i> debe ser capaz de seleccionar los valores de flujo no usados para no solicitar un valor específico ya usado.</p>	<p>Los paquetes que no pertenecen a ninguno de los flujos que tienen reserva de recursos no reciben ningún tratamiento especial.</p>
<p>Una fuente debe asegurar que no hace re-uso no intencional de los valores de <i>Etiqueta de Flujo</i> que se están usando en el momento. Los valores de <i>etiquetas de flujo</i> usados con un par específico de <i>direcciones fuente y destino</i> no deben ser asignados a nuevos flujos con el mismo par de direcciones dentro de los 120 segundos de la terminación del flujo previo. El <i>nodo fuente</i> debe proveer los medios para que las aplicaciones y los protocolos de transporte especifiquen períodos de cuarentena más largos que el valor por defecto de 120 segundos para flujos individuales.</p>	<p>En <i>IntServ6</i>, los valores de las etiquetas de flujo se calculan con base en la <i>Quíntupla</i>, lo que hace que se tome en cuenta la aplicación y las conexiones de transporte en este cálculo.</p>
<p>Para evitar re-uso accidental, el <i>nodo fuente</i> debería seleccionar los nuevos valores de <i>Etiqueta de Flujo</i> en una secuencia bien definida (e.g. <i>secuencial o pseudo-aleatoria</i>) y usar un valor inicial que evite el re-uso de valores de <i>Etiqueta de Flujo</i> cada vez que el sistema se re-inicia. El valor inicial debe ser derivado de un valor previo almacenado en una memoria no-volátil, o en ausencia de tal historia, debe usarse un valor inicial generado aleatoriamente usando técnicas que producen unas buenas propiedades de aleatoriedad.</p>	<p>En <i>IntServ6</i>, podría ocurrir que un <i>nodo fuente</i> asigne dos valores de <i>Etiqueta de Flujo</i> iguales. Esto no constituye un problema ya que se resuelve en los <i>Encaminadores</i> mediante la <i>tabla de resolución de Colisiones</i>.</p>
<p>Este requerimiento no tiene efecto en la propuesta <i>IntServ6</i> pues no se utilizan las <i>direcciones Origen y Destino</i> para clasificar los flujos. Sin embargo, la asignación de la <i>Etiqueta de Flujo</i> a partir de la <i>Quíntupla</i> y una <i>función hash</i> hace que la probabilidad de que se repitan <i>números hash</i> con iguales direcciones origen y destino sea muy baja.</p>	<p>Si ocurre una colisión de valores hash, esto se soluciona en los <i>Encaminadores</i> mediante la <i>tabla de resolución de colisiones</i>. Esto trae la ventaja de no tener restricciones de tiempo entre <i>tripletas</i> ni en los valores iniciales de las etiquetas al re-iniciar el sistema.</p>

Requerimiento del RFC 3697	Cumplimiento en IntServ6
<p>Para habilitar el tratamiento específico por flujo, se requiere establecer el estado del flujo en todos o en un subconjunto de los nodos IPv6 en el camino desde la fuente hasta el destino. Los métodos para el establecimiento del estado, también como los modelos para el tratamiento de un flujo específico serán definidos en especificaciones separadas.</p>	<p>Todos estos aspectos son cubiertos en IntServ6 debido al uso del protocolo RSVP para la el establecimiento de las reservas y la gestión del estado de las mismas. RSVP posee mecanismos como la necesidad de refresco de las reservas que permite recuperaciones de cambios en el camino de los paquetes.</p>
<p>Para habilitar la co-existencia de diferentes métodos en nodos IPv6, los métodos deben alcanzar los siguientes requisitos básicos:</p>	<p>En IntServ6 los Encaminadores tienen sus mecanismos de Control de Admisión que evitan que estos se comprometan a más recursos de los que pueden ofrecer.</p>
<ol style="list-style-type: none"> a. El método debe proveer los medios para la limpieza del estado del flujo en los nodos IPv6 que proveen tratamiento específico por flujos. Los métodos de señalización donde el nodo fuente está involucrado son libres de especificar tiempos de vida del estado del flujo mayores que el valor por defecto de 120 segundos. b. Los métodos de establecimiento de los flujos deben ser capaces de recuperarse del caso donde el estado del flujo requerido no pueda ser soportado. 	

3.8 Conclusiones

En este capítulo se planteó una solución para soportar *Calidad de Servicio* en Internet que opera sobre redes fijas y móviles basadas en el protocolo IPv6. Esta solución está fundamentada en la *Arquitectura de Servicios Integrados* y utiliza el campo de *Etiqueta de Flujo* de la cabecera IPv6 para identificar los flujos que tienen reservas de recursos en los *Encaminadores*. La utilización del campo *Etiqueta de Flujo* de IPv6 no ha sido estandarizada, existiendo hasta el momento sólo ciertas especificaciones de uso y algunas propuestas de su utilización que también fueron estudiadas en esta sección. Tales propuestas sólo abarcan algunos aspectos de una posible solución pero no plantean soluciones completas. Estos aspectos son entre otros, el formato de interpretación de la *Etiqueta de Flujo* y algunas definiciones de arquitecturas de *QoS* descritas de forma muy general.

Debido a que IntServ6 utiliza las *Etiquetas de Flujo* de IPv6 para realizar un proceso cercano al de *Conmutación de Etiquetas (Tagging)* y a su vez utiliza IntServ para soportar *Calidad de Servicio*, esta tecnología presenta un desempeño híbrido similar al de MPLS en algunos aspectos y similar al desempeño de IntServ en otros. Cabe aclarar que MPLS es una solución que requiere de *Encaminadores* especiales que realizan la *Conmutación de Etiquetas*, de forma que para agilizar el Encaminamiento de los paquetes IP, se requiere de una *Red de Transporte MPLS* que lleve los paquetes desde

una red *IP* origen hasta una red *IP* destino. Una ventaja que ofrece *MPLS*, además de que es un medio de transporte rápido, es que funciona tanto con paquetes *IPv4* como con paquetes *IPv6*, a la vez que permite hacer Encaminamiento con restricciones de *Calidad de Servicio*, para lo que usa protocolos de señalización como *CR-LDP* (*Constraint Routing Label Distribution Protocol*) o *RSVP-TE* (*RSVP with Traffic Engineering Extension*) para establecer el camino adecuado a las especificaciones de *QoS*. Estos protocolos en sí no permiten a *MPLS* ofrecer *QoS*, sólo son usados para establecer el mejor camino con base en caminos con una *Calidad de Servicio* ya existente.

Por otro lado, *IntServ6* ofrece una solución que permite ofrecer *QoS* en redes *IPv6* con una mejoría considerable en el tiempo de procesamiento de los *Encaminadores* con respecto a *IntServ*. Además, *IntServ6* es una solución para *Redes de Acceso* que se puede interconectar fácilmente en un *Núcleo MPLS* debido al uso de etiquetas para la clasificación de los paquetes.

Una característica importante de *IntServ6* es que ofrece soluciones simples y eficientes a problemas tales como el *Soporte de Reservas dentro de Túneles* y la *Agregación de Flujos*. Para ambos procesos *IntServ6* utiliza un procedimiento similar al *Label Stacking* de *MPLS*. Además, logra solucionar el problema de *Reservas Dentro de Túneles* a nivel de *capa 3* (*Internet*) mientras que la solución estándar de *IntServ* debe acudir a información de la *capa 4* (*Transporte*). Por otra parte, la solución de la *Agregación de Flujos* de *IntServ6* permite reducir el problema de escalabilidad que ha presentado la solución *IntServ* estándar.

Adicionalmente, *IntServ6* posee extensiones para operar sobre redes móviles. Tales extensiones están diseñadas para operar con los protocolos de movilidad existentes de forma que no requieran la alteración de los dichos protocolos. Aunque *IntServ6* podría operar con el protocolo *Mobile IPv6*, es preferible su operación sobre *HMIPv6* para reducir el tiempo de *Traspaso*.

Finalmente, es importante resaltar que *IntServ6* cumple con todas las recomendaciones especificadas por la IETF en el *RFC 3697* [61].

4 Evaluación de IntServ6

En el presente capítulo se estudian diferentes aspectos de *IntServ6* que permiten demostrar que dicha propuesta presenta una serie de características que mejoran el comportamiento de los *Encaminadores* con respecto a la tecnología estándar *IntServ* en diferentes situaciones. Es así que este estudio se inicia en la sección 4.1 con la descripción de la plataforma de evaluación utilizada. En la sección 4.2 se realiza un modelado y análisis del desempeño de los *Encaminadores* en aspectos como el *ancho de banda de memoria* requerido y el rendimiento con respecto al *número de hilos* utilizados en el procesamiento. En la sección 4.3 se presenta un modelo del *retardo medio* de los paquetes, mientras que en la sección 4.4 se muestra el comportamiento del *tráfico servido* por los *Encaminadores* tanto en *reservas simples* como en *reservas dentro de túneles*. En la sección 4.5 se estudia la dependencia del retardo de los paquetes con respecto a la movilidad de los usuarios en *Encaminadores de Acceso Inalámbricos*. Finalmente en la sección 4.6 se hace un análisis del *tiempo de Traspaso* y en la sección 4.7 se hace un análisis de la gestión de etiquetas de *IntServ6* en comparación con *MPLS* cuando ambas operan sobre redes móviles de acceso.

4.1 El procesador de red IXP 1200 de Intel como plataforma de Evaluación

Tradicionalmente, los *Routers* de alto rendimiento han sido diseñados usando Circuitos Integrados de Aplicación Específica (*ASIC, Application Specific Integrated Circuits*) que procesan paquetes a altas velocidades. Generalmente estos *ASICs* tienen procesos de diseño y desarrollo largos y son difíciles de actualizar para agregar nuevas funcionalidades. Para permitir el rápido desarrollo de servicios de red nuevos, los *Routers* de próxima generación han sido construidos usando procesadores de red programables que soportan la programación de funciones de procesamiento de paquetes mediante *software* [34].

El procesador de red *IXP1200* de *Intel* fue diseñado para construir *Routers* con enlaces *OC-3*, es decir, con anchos de banda de 100Mbps a 500Mbps, mientras que los

procesadores de las familias *IXP24XX* y *IXP28XX* permiten anchos de banda de hasta 10Gbps (enlaces *OC-192*). Sin embargo, estos últimos al igual que dispositivos de otros fabricantes (p.ej. *Clearwater CNP810SP*, *IBM Rainer*, *MMC nP7510*, *Motorola C-5*) utilizan variantes de la técnica de *Multi-Threading* (uso de múltiples hilos para el procesamiento) que la familia *IXP1200* [74]. Por tanto, al usar el *IXP1200* como base para el análisis de rendimiento de diferentes tecnologías de conmutación de paquetes, se obtendrán resultados con comportamientos que permiten discutir las implicaciones sobre las arquitecturas de *Routers* de próxima generación [34].

En esta tesis se ha utilizado el procesador de red programable *IXP1200* de *Intel* para realizar la evaluación de la propuesta. La utilización de este dispositivo está justificada además por su extenso uso como procesador de red en diferentes tipos de conmutadores *LAN* y *WAN* con tecnologías *IntServ*, *DiffServ* y *MPLS* [75, 76]. Este dispositivo ha sido ampliamente usado en educación e investigación en diferentes universidades tales como Princeton [77, 78], Universidad de California en Berkeley [79], la Universidad de Massachussets [80] y también en el *Programa de Educación en Innovación* de *Intel* en países como India [81].

Finalmente, este procesador ha sido utilizado en diferentes investigaciones [33, 34, 74, 82] para evaluar varias clases de *Routers* tales como *Routers IP* básicos, *Routers MPLS* y *Routers* con tecnologías para el soporte de *QoS* como *IntServ* y *DiffServ*. Los modelos obtenidos en [33, 34] fueron utilizados en esta tesis para obtener un modelo del rendimiento de diferentes tipos de *Routers* y también se usaron los parámetros del procesador *IXP1200* para obtener modelos de evaluación de otros parámetros de *QoS*. A continuación se describirá la arquitectura básica del procesador de red *IXP1200*. Para mayor información, referirse a [83].

4.1.1 El procesador IXP1200

El procesador de red *IXP1200* [83] (ver Figura 4.1) contiene 4 puertos *MAC*, un procesador de núcleo *StrongARM*, seis *CPUs* tipo *RISC* (conocidas como *Micro-máquinas*), un controlador de bus propietario (bus *IX* de 64 bits a 66Mhz), un

controlador *PCI*, unidades de control para acceso a las memorias *SRAM* y *DRAM* que se encuentran fuera del circuito integrado, y una pequeña cantidad de memoria temporal interna (*Scratch Pad*). El núcleo *StrongARM* se usa para el procesamiento de caminos de control tales como paquetes de excepción de camino lento, gestión de *Tablas de Encaminamiento* y otra información del estado de la red. Las *Micro-máquinas*, por otro lado, son usadas para el procesamiento del camino de los paquetes; estas procesan múltiples paquetes en paralelo. Cada *Micro-máquina* está asociada con una *Memoria Caché* de instrucciones de 4KB. Tanto el núcleo *StrongARM* como las *Micro-máquinas* tienen un reloj de 200 Mhz.

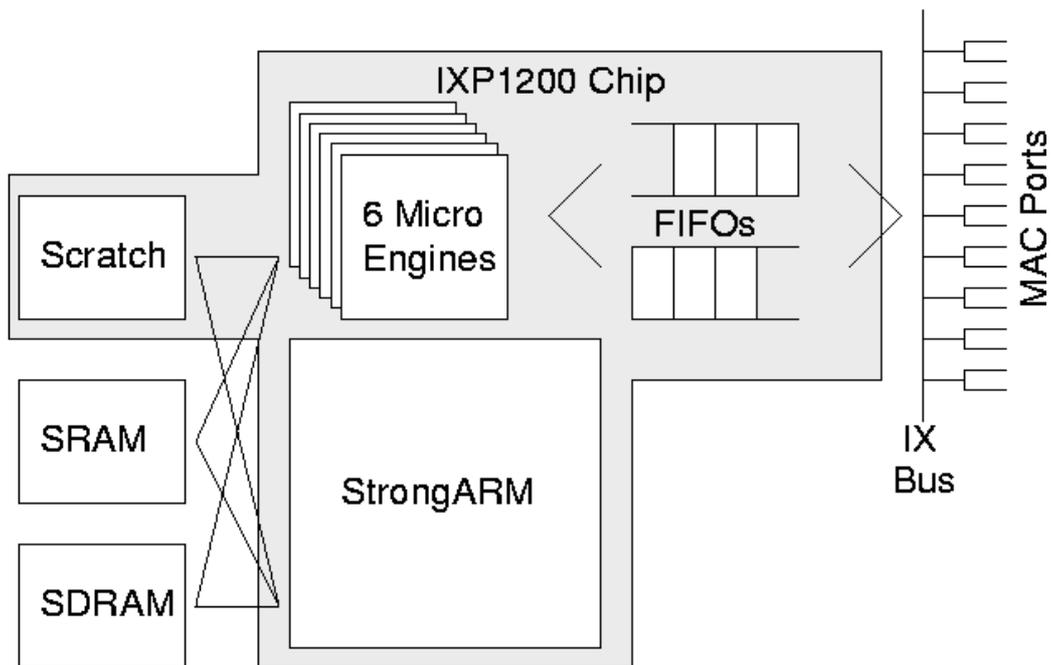


Figura 4.1 Diagrama de Bloques del sistema *IXP1200* [83].

Para que el procesador de red esté capacitado para trabajar a velocidades de línea, es importante esconder el retraso en que se incurre mientras se accede a la memoria durante el procesamiento del paquete. Para obtener esto, cada *Micro-máquina* tiene 4 hilos; una *Micro-máquina* puede conmutar para cambiar de contexto desde un hilo a otro en un solo ciclo. La adquisición de una ranura (recurso de procesamiento) se consigue en el *IXP1200* usando una cola única en la cual se colocan las solicitudes para cualquier ranura, así esté disponible o no. Las solicitudes de ranura son servidas en orden *FIFO* desde esta cola, que podría agregar retardos debido al bloqueo de la

cabecera de la cola. Aunque no se requiere explícitamente, el uso más natural de la memoria *DRAM* es almacenar paquetes. Este almacenamiento opera en función del tamaño de la memoria (256MB para el sistema de evaluación utilizado en [33]) y la velocidad del acceso a memoria (33-34 ciclos). La memoria *SRAM* es el sitio natural para almacenar información de control frecuentemente almacenada, tal como información de Encaminamiento, estado por flujos, etc. La *SRAM* es relativamente pequeña en tamaño (8MB para los trabajos en [33]) y tiene un tiempo de acceso más pequeño (16-20 ciclos). La memoria temporal interna del circuito integrado se utiliza para leer y escribir mensajes de control cortos y datos que son compartidos entre las *Micro-máquinas* y el núcleo *StrongARM*.

4.1.2 Arquitectura de Software del IXP1200

El procesamiento de paquetes en la plataforma optimizada de *E/S* del Router *IXP1200* está dividida en dos fases: La fase de *Recepción* y la fase de *Transmisión* [33]. Cada hilo es asignado estadísticamente a un puerto –los hilos asignados a los puertos de entrada ejecutan la fase de *Recepción* y los hilos asignados a los puertos de salida ejecutan la fase de *Transmisión*. Un hilo receptor transfiere un paquete entrante a la *DRAM*, luego procesa el paquete, identifica el puerto saliente (ya sea a través del *Estado por Flujo* o la *Tabla de Encaminamiento*), y coloca el descriptor del paquete en la cola (podría ser una cola de prioridades) que corresponde al puerto de salida. Cuando un puerto de salida está disponible para la *Transmisión*, el hilo transmisor correspondiente saca el descriptor del paquete de la cola de ese puerto, y transfiere el paquete correspondiente desde la *DRAM* al puerto.

Debido a la limitada cantidad de instrucciones disponibles en la *Memoria Caché* para cada *Micro-máquina*, es deseable ejecutar el mismo código en todos los hilos de una misma *Micro-máquina*. Adicionalmente, cuando diferentes hilos dentro de la misma *Micro-máquina* ejecutan código similar, sus patrones de acceso a memoria son similares, y los accesos a memoria pueden ser intercalados eficientemente. En consecuencia, los retardos de memoria pueden ser escondidos efectivamente. Así, cada una de las 6 *Micro-máquinas* pueden ser receptoras o transmisoras, con todos los 4 hilos

en ella ejecutando la fase de *Recepción* o de *Transmisión* respectivamente. Los 4 hilos en cada *Micro-máquina* receptora (o transmisora) son asignados estadísticamente a los 4 puertos de entrada (o salida).

4.2 Mejora en el Rendimiento de los Encaminadores

Debido a que *IntServ6* utiliza las *Etiquetas de Flujo* de *IPv6* para realizar un proceso cercano al de *Conmutación de Etiquetas (Tagging)* y a su vez utiliza *IntServ* para soportar *Calidad de Servicio*, esta tecnología presenta un desempeño híbrido similar al de *MPLS* en algunos aspectos y similar al desempeño de *IntServ* en otros.

Cabe aclarar que *MPLS* es una solución que requiere de *Encaminadores* especiales que realizan la *Conmutación de Etiquetas*, de forma que para agilizar el Encaminamiento de los paquetes *IP*, se requiere de una red de transporte *MPLS* que lleve los paquetes desde una red *IP* origen hasta una red *IP* destino. La ventaja que supone esta solución, además de que es un medio de transporte rápido, es que funciona tanto con paquetes *IPv4* como con paquetes *IPv6*, a la vez que permite hacer Encaminamiento con restricciones de *Calidad de Servicio*, para lo que usa protocolos de señalización como *CR-LDP (Constraint Routing Label Distribution Protocol)* [28] o *RSVP-TE (RSVP with Traffic Engineering Extension)* [84] para establecer el camino adecuado a las restricciones de *QoS*. Estos protocolos en sí no permiten a *MPLS* ofrecer *QoS*, sólo son usados para establecer el mejor camino con base en caminos con una *Calidad de Servicio* ya existente.

Por otro lado, *IntServ6* ofrece una solución intermedia que permite ofrecer *QoS* en redes *IPv6* con una mejora en el tiempo de procesamiento de los *Encaminadores* con respecto a *IntServ*. Además, es una solución de *red de acceso* que se puede mapear fácilmente en un *Núcleo MPLS* ya que sólo requiere un intercambio de etiquetas en los *Encaminadores Frontera*.

La evaluación de *IntServ6* en el aspecto que se refiere al desempeño de los *Encaminadores* [7-9], está basada en el estudio desarrollado por B. Hardekopf en [34] y en el estudio realizado por J. Kaur [35]. Esta sección describe un modelo del

rendimiento de un *Encaminador* basado en el procesador *IXP1200* de *Intel*. El modelo obtenido en [34] comprende tres tareas básicas de un *Encaminador*: *Clasificación de Paquetes*, *Selección de la Ruta* y *Planificación de Paquetes*. Para cada una de las tareas se encuentra un modelo que es genérico y que puede adaptarse para dos disciplinas diferentes. La idea básica es que, mediante la combinación de las diferentes disciplinas en los tres procesos mencionados, se puede obtener un modelo simplificado de diferentes *Arquitecturas de Servicio* entre las que están: *IP*, *IntServ* y *MPLS*. En el caso de la *Clasificación de Paquetes* se estudian las disciplinas *Tagging* y *Matching*. La primera es el equivalente de la *Conmutación de Etiquetas*, mientras que la segunda es el equivalente del *Hashing*. El modelo encontrado para la *Clasificación de Paquetes* es aplicable a los dos tipos de disciplinas, *Tagging* y *Matching*, la diferencia entre una y otra es el valor del parámetro k , que se explica más adelante.

A continuación se describirán los modelos obtenidos por Hardekopf para el caso de la *Clasificación de Paquetes* y luego se explicará una modificación realizada a estos para incluir el efecto de la *Resolución de Colisiones*, pues este no se contempla en el modelo original.

4.2.1 Modelo de Hardekopf

Los parámetros que se modelan son:

- El *Ancho de Banda de Memoria* (BW_m), el cual está dado en bits por segundo (bps) y representa el número de bits que son leídos de la memoria por el procesador durante una tarea.
- El *Número de Hilos* (H) necesario en el procesador para realizar las tareas y cumplir con el re-envío de paquetes a la velocidad del enlace de entrada.

Los modelos desarrollados por Hardekopf para BW_m y H se encuentran en las ecuaciones 4.1 y 4.2 respectivamente:

Ecuación 4.1:

$$BW_m = \frac{(8k+2) \times 32C}{M}$$

Ecuación 4.2:

$$H = \frac{((k+1) * L_{SRAM} + L_{SDRAM}) * C}{M}$$

La Ecuación 4.1 es una multiplicación del número de referencias a memoria ($8k+2$), el tamaño de la palabra (32 bits) y el número de paquetes por segundo (C/M). La descripción detallada de cada una de las partes de esta ecuación es la siguiente:

- El número 8 es el tamaño de la fila de la *Tabla Hash*. Está dado en *palabras (words)*.
- La variable k es el número máximo de lecturas de la *Tabla Hash* que toma el proceso. Más adelante se explica en detalle este parámetro.
- El número 2 es el tamaño de la fila de un estado del flujo (dado en *palabras*).
- El número 32 es el tamaño de la *palabra* del procesador (*bits/palabra*).
- La variable C es la velocidad del enlace entrante (dada en *bps*).
- La variable M es el tamaño del paquete (dado en bits).

Para entender mejor cómo se obtiene la Ecuación 4.1, se debe explicar la situación a modelar (ver Figura 4.2). Esta consiste en que se tiene una *Tabla Hash* en la que se encuentran ubicadas las entradas (apuntadores) a los recursos de los flujos en la *Tabla de Reservas* (que contiene el *Estado de la Reserva del Flujo* y el *Descriptor del Flujo*). Estas entradas son denominadas *Ranuras Hash*. Cada vez que un paquete es procesado por un hilo para obtener el *Identificador del Flujo (Número Hash)*, debe accederse esta tabla para leer el estado del flujo y modificar el *Descriptor de Flujo*, específicamente los parámetros *Tiempo Máximo de Retardo* (o *Deadline*) y *Número de Paquetes por Flujo*. El proceso de modificación del *Descriptor de Flujo* requiere de un aseguramiento del recurso durante un tiempo para evitar que otros hilos modifiquen estos datos mientras se realiza este paso. Es por esto que se produce una *Colisión*

cuando dos o más hilos requieren acceder una misma *Ranura*. En el peor de los casos, si hubiese k hilos, todos los k hilos tendrán paquetes con el mismo *Número Hash*. En este momento debe resolverse la *colisión* y por tanto el paquete al que le corresponda esperar más, deberá esperar $k-1$ accesos a la *Tabla Hash* por parte de los otros $k-1$ hilos. Por tanto, hará su acceso en el k -ésimo intento y así se deben considerar k lecturas de la *Tabla Hash*, además de un acceso para modificar el *descriptor de flujo*, por lo que el número total de accesos a memoria será de $k+1$.

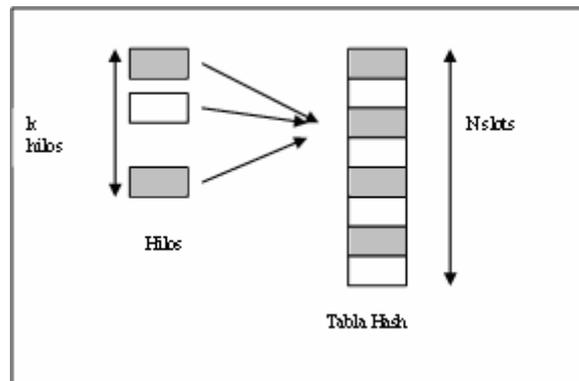


Figura 4.2 Situación analizada en el modelo original

Bajo las condiciones mencionadas anteriormente, el número de colisiones es igual al número de hilos del *Encaminador* y en la medida que aumente k , aumentará el tiempo de procesamiento del paquete. Este valor de k es 1 para el caso del *Tagging* debido a que el proceso de cálculo del *número hash* no existe en este tipo de clasificación por lo que el paquete pasa directamente al paso de revisar la *Tabla de Estado del Flujo* y modificar el *Descriptor de Flujo*.

La Ecuación 4.2 por su parte, se determinó teniendo en cuenta que por cada paquete se deben realizar las siguientes operaciones:

- Una lectura de *SDRAM* para obtener el paquete
- En el peor de los casos, se obtendrá la lectura de la fila de la *Tabla Hash* (ubicada en *SRAM*) en el intento k para encontrar la entrada apropiada en esta tabla.

- A continuación debe hacerse una escritura en *SRAM* para actualizar el estado del flujo (*Flow Descriptor*).

Por tanto, la clasificación ocupa 1 hilo para un promedio de: $(k+1)$ referencias a *SRAM* por paquete y 1 referencia a *SDRAM* por paquete. Las variables L_{SRAM} y L_{SDRAM} son los tiempos de lectura de *SRAM* y *SDRAM* respectivamente.

Resumiendo, la Ecuación 4.1 y la Ecuación 4.2 son modelos generales que se pueden aplicar tanto al caso de *Tagging* (haciendo $k=1$) como al caso de *Matching* (haciendo $k>1$).

4.2.2 Mejoras al Modelo de Hardekopf

El modelo de Hardekopf no ha tenido en cuenta el efecto de la resolución de las colisiones cuando dos flujos diferentes tienen un mismo *Número Hash* durante el proceso de *Matching*. En nuestra propuesta hemos introducido este aspecto y, ya que deben ser comparadas las tres tecnologías (*MPLS*, *IntServ* e *IntServ6*) bajo las mismas condiciones, debemos introducir este efecto en el modelo de Hardekopf.

El efecto de las colisiones se introduce agregando la lectura de la fila de la *tabla de colisiones* a la Ecuación 4.1 y la Ecuación 4.2. Así, la Ecuación 4.1 quedaría así:

Ecuación 4.3:

$$BW_m = \frac{32C * (8k + 2 + 6mC_h)}{M}$$

Podemos observar que se ha agregado el término $6mC_h$, el cual está comprendido por mC_h , que es el tamaño de la *Tabla de Resolución de Colisiones* y el número 6. Es decir, en el peor caso, el flujo buscado en esta tabla estará ubicado en el último lugar de esta tabla y por tanto se harán mC_h lecturas de ella, donde m es el número de flujos ubicados en la *Tabla Hash* y C_h es la tasa de *colisiones hash* de estos m flujos. Además se deberá leer cada una de las filas en la búsqueda, lo que hace que se deban leer 6

palabras por fila. Este número (6) es calculado con base en los componentes de una fila de la tabla (número hash de 32 bits, *Quíntupla* de 104 bits para el caso de *IPv4* y el apuntador al estado de la reserva que es de 32 bits).

El modelo modificado para la Ecuación 4.2 quedaría entonces representado introduciendo mC_h lecturas a la *Tabla de resolución de Colisiones*. Por tanto, la ecuación (4.2) queda así:

Ecuación 4.4:

$$H = \frac{C * ((k + 1 + mC_h) * L_{sram} + L_{sdram})}{M}$$

4.2.3 Modelo de *IntServ6*

El modelo de *IntServ6* para el *Número de Hilos (H)* necesario es idéntico a la Ecuación 4.4, mientras que nuestro modelo para el *Ancho de Banda de Memoria* toma en cuenta las colisiones por lo que es similar al de la Ecuación 4.3 y se observa en la Ecuación 4.5,

Ecuación 4.5:

$$BW_m = \frac{32C * (1k + 2 + 11mC_h)}{M}$$

Se puede observar en esta ecuación que se hacen ciertas modificaciones al tamaño de la fila de la *Tabla Hash*. En nuestra propuesta la *Tabla Hash* sólo se compone del campo *Colisión* (1bit) y el campo *Estado de Reserva* (20 bits), ya que la *Etiqueta de Flujo* (*número Hash*) sirve como índice y no forma parte de la tabla. Se debe aclarar que en la Figura 3.3 (capítulo 3), la *Etiqueta de Flujo* se ha colocado como campo por facilidad de la explicación, pero realmente no es parte de la tabla. Por tanto, el número de palabras necesarias para leer esta fila de la *Tabla Hash* es de 1 palabra de 32 bits. Adicionalmente, al modelo representado por la Ecuación 4.5 se le han colocado filas de 11 palabras en la *Tabla de Resolución de Colisiones*. Esto se debe a que se manejan los

campos: *Etiqueta de Flujo* (Número Hash de 20 bits), la *Quíntupla* (296 bits para IPv6) y la *Identificación del estado de la Reserva* (20 bits).

Es importante aclarar que en la Ecuación 4.5 se debe hacer $k=1$, ya que el procedimiento de búsqueda en la *Tabla Hash* es el mismo del *Tagging*. Adicionalmente, el valor para la *tasa de colisiones* se puede tomar como el número de colisiones que pueden ocurrir para una *función Hash* perfecta que distribuye uniformemente los *números Hash* [85]. Esta expresión es la siguiente:

Ecuación 4.6:

$$C_h = 1 - \frac{N * \left(1 - \left(\frac{N-1}{N} \right)^m \right)}{m}$$

Donde N es el tamaño de la *Tabla Hash*; y m es el número de flujos que han sido ubicados en la *Tabla Hash*. Para obtener las expresiones completas, el valor de C_h se reemplaza en las expresiones comprendidas entre la Ecuación 4.3 a la Ecuación 4.5.

4.2.4 Resultados Obtenidos

Se analizaron tres situaciones bajo diferentes condiciones, teniendo en cuenta las características de cada una de las arquitecturas. Como se observa en la Tabla 4.1, la arquitectura *IntServ* tiene como peor caso k lecturas en la *Tabla Hash* debido a que usa el *Matching* como método de clasificación. También se observa que *MPLS* e *IntServ6* sólo hacen una lectura debido a que se utiliza el valor de la etiqueta como índice de búsqueda en la *Tabla de Encaminamiento (Tagging)*. Por otro lado, se puede ver que *MPLS* tiene un $C_h=0$ debido a que no presenta colisiones, mientras que *IntServ* e *IntServ6* sí presentan colisiones porque usan un número *Hash* para identificar los flujos. El análisis realizado y sus resultados se explican a continuación.

Tabla 4.1 Características de las arquitecturas en el modelo general

Arquitectura	Número de lecturas en la tabla hash (k)	Tamaño tabla de resolución de colisiones
<i>IntServ</i>	>1	mC_h
<i>MPLS</i>	1	0
<i>IntServ6</i>	1	mC_h

4.2.4.1 Relación Ancho de banda de memoria (BW_m)- lecturas tabla hash (k)

Se graficó el ancho de banda de memoria y se varió k , manteniendo constante el número de flujos (m). La velocidad del enlace de entrada es de $C=400\text{Mbps}$ y el tamaño de los paquetes es de 512 bits. El tamaño de los paquetes se dejó constante en este valor debido a que es el tamaño de paquete más pequeño posible, por lo que la tasa de paquetes que llegan al *Encaminador* a su vez será la mayor posible, permitiendo de esta manera evaluar el *Encaminador* bajo las peores condiciones. La Figura 4.3 muestra el *Ancho de Banda de Memoria* normalizado, por lo que se observa la variación de este en términos de porcentaje de crecimiento. Estas condiciones afectan a *IntServ*, pues el aumento del número de lecturas de la *Tabla Hash* se debe a que ha aumentado el número de hilos en el mismo número. Esto trae como consecuencia que el *Ancho de banda de memoria* para *IntServ* crezca proporcionalmente con el *Número de Hilos*. Esta variación del *Número de Hilos* no afecta a *MPLS* ni a *IntServ6*, pues el número de lecturas que se realiza en la tabla es 1 ($k=1$) debido al uso del *Tagging*, por lo que se mantiene constante el *Ancho de Banda de Memoria* para estas dos arquitecturas.

La cantidad de *Colisiones Hash*, es decir, la asignación del mismo *Número Hash* a dos flujos diferentes se mantiene constante en este caso ya que el tamaño de la *Tabla Hash* en *IntServ6* es constante y de valor 2^{20} (debido a que la *Etiqueta de Flujo* tiene 20 bits de longitud) y además se mantiene constante en 6400 el número de flujos (m) para esta simulación.

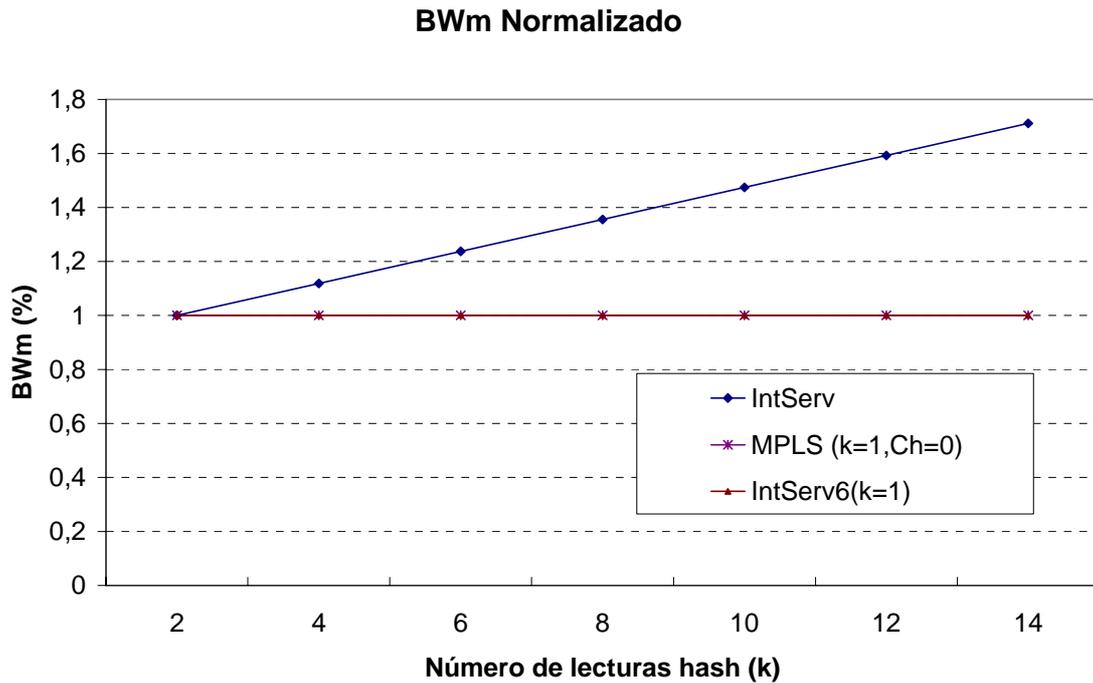


Figura 4.3 Variación del *Ancho de Banda de Memoria* con el *Número de Lecturas Hash*

4.2.4.2 Relación *Ancho de Banda de memoria (BW_m)*- *Número de Flujos*

Los resultados se observan en la Figura 4.4 y la Figura 4.5; el *Ancho de Banda de Memoria* se graficó bajo las siguientes condiciones:

- Se mantuvo constante el tamaño de la *Tabla Hash* en 2^{20}
- Se varió la *Cantidad de Flujos (m)*
- La capacidad del enlace de entrada se mantuvo en $C=400\text{Mbps}$ (máxima capacidad del IXP1200)
- El tamaño de los paquetes fue de 512 bits
- k tiene un valor de 4 (*Número de Hilos* del IXP1200)

En este caso se observa que el *Ancho de Banda de Memoria* permanece constante para *MPLS*, mientras que para *IntServ* e *IntServ6* presenta un aumento en la medida que aumenta el *Número de Flujos*. Este aumento se debe a que al incrementar el número de flujos, aumentan las *Colisiones Hash* debidas a la asignación del mismo *Número Hash* a

diferentes flujos. Obsérvese que el *Ancho de Banda de Memoria* de *IntServ6* tiene un mejor comportamiento con respecto a *IntServ* para pocos flujos, mientras que el comportamiento de esta misma variable en *IntServ6* para un *Número de Flujos* superior a los 4000 aproximadamente es igual al de *IntServ* pero con un valor un poco más alto. Esto no constituye un problema para *IntServ6* en *Redes de Acceso* donde el *Número de Flujos* está por debajo de este valor. Además, el mayor aumento del *Ancho de Banda de Memoria* en *IntServ6* con respecto a *IntServ* para más de 4000 flujos se debe a que el número de bits a almacenar en *IntServ6* es mayor a causa del uso de direcciones *IPv6* de 128 bits. Este aspecto es solucionable en la medida que la capacidad de almacenamiento de las memorias de semiconductores aumenta y el tamaño de la palabra de los procesadores crece. Nótese también que en la gráfica el crecimiento parece proporcional, pero esto es un efecto de haber elegido potencias de dos para los valores del número de flujos. En realidad el crecimiento es exponencial (en potencias de dos), como se observa en la Figura 4.4 y en la Figura 4.5, cuya diferencia está en la escala que se maneja (variaciones en unidades de 1000 flujos para la primera y de 10000 flujos para la segunda).

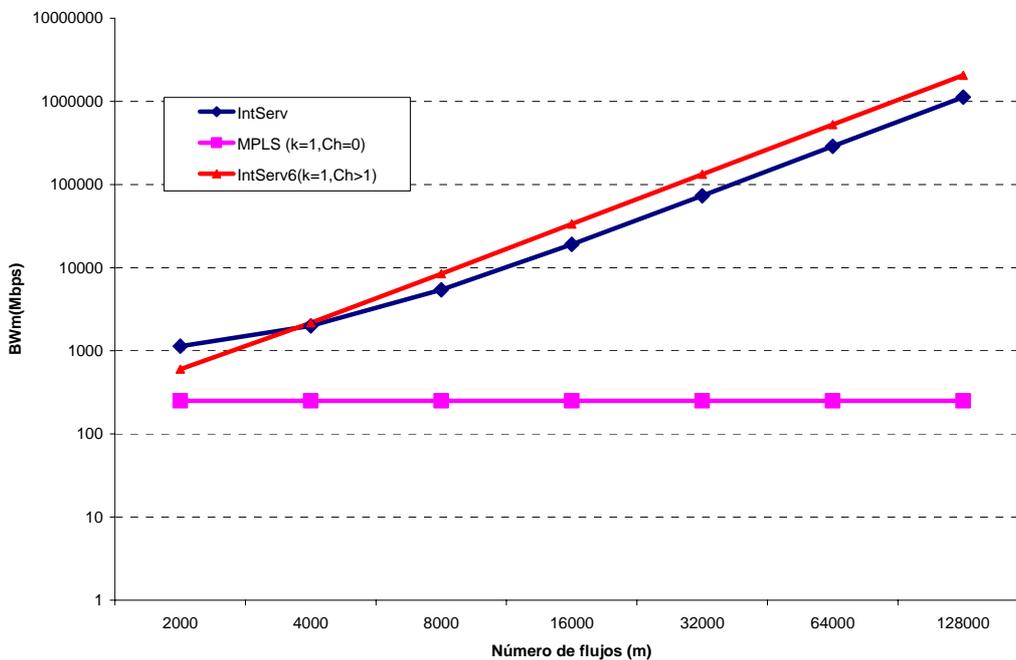


Figura 4.4 Variación *Ancho de banda de memoria* con respecto al *número de flujos*.

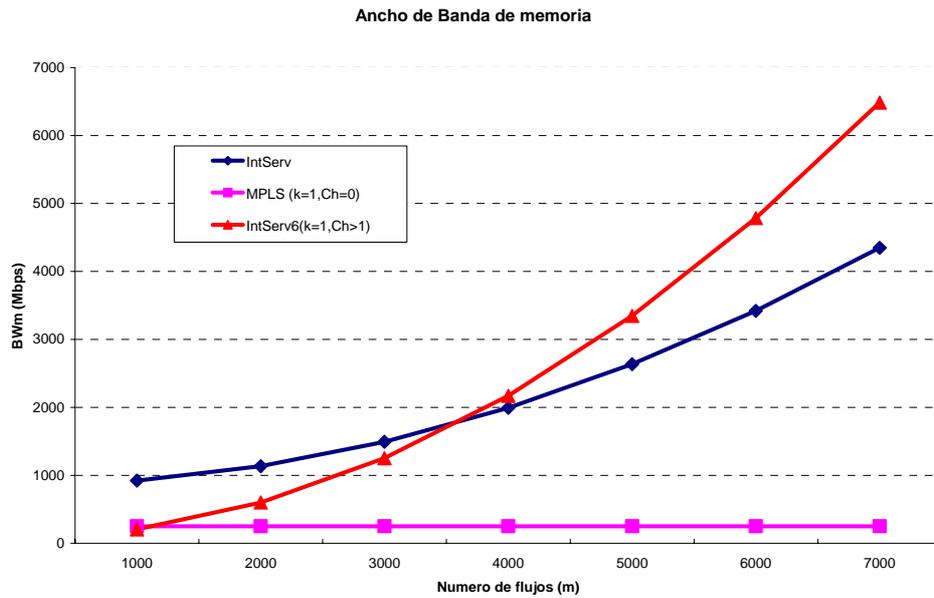


Figura 4.5 Variación del *Ancho de banda de memoria* con respecto al *número de flujos* (escala: 1000)

4.1.1.1. *Relación del Throughput con el número de hilos del Encaminador*

Se simuló el *Throughput* del proceso de clasificación de flujos del *Encaminador*, es decir, cuántos bits por segundo puede procesar para cada una de las tecnologías cuando se varía el número de hilos del procesamiento. Los resultados se observan en la Figura 4.6. Si se tiene en cuenta que este *Encaminador* puede trabajar con enlaces de hasta 400Mbps, las tres tecnologías pueden trabajar a esa velocidad sin problemas. Sin embargo, al aumentar el número de hilos el comportamiento de *MPLS* es el mejor seguido por *IntServ6*, mientras que *IntServ* presenta el peor de los comportamientos entre las tres tecnologías. Este aspecto es una ventaja ya que al incrementarse las prestaciones del procesador nuestra propuesta mejorará aún más su desempeño respecto al de *IntServ*.

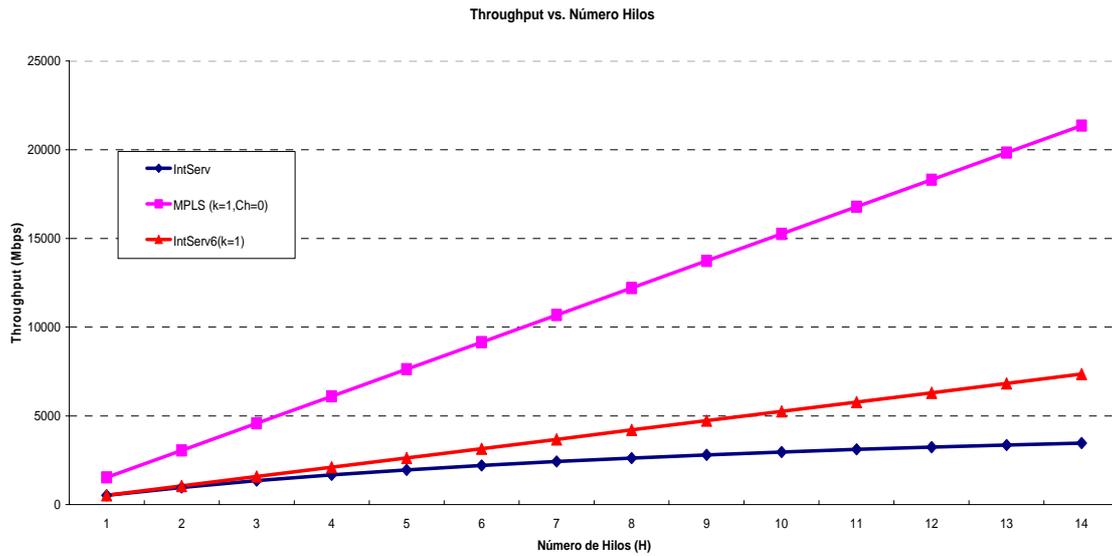


Figura 4.6 *Throughput* del proceso de clasificación del *Encaminador*

4.3 Evaluación del Retardo Medio de los paquetes

En esta sección se estudia el retardo medio de los paquetes en un *Encaminador de Servicios Integrados* [6] y se compararán las dos tecnologías objeto de nuestro estudio: *IntServ* e *IntServ6*. El *Retardo Medio* de los paquetes es un parámetro de gran importancia para cierto tipo de aplicaciones *Semi-Elasticas* tales como la transferencia de ficheros, en donde el retardo medio de los paquetes determina el *Tiempo Medio* que durará la descarga de tal fichero, pero no es necesario tener un límite drástico en el tiempo de retardo de cada paquete o en la variación del retardo de los mismos. Igualmente, para aplicaciones de *Playback*, es decir, aplicaciones tales como las que permiten escuchar emisoras de radio y que utilizan *Buffers* de almacenamiento, el *Retardo Medio* de los paquetes y la varación del caudal de los paquetes podrían servir para dar una idea del *Tamaño* del *Buffer* de almacenamiento de los paquetes.

4.3.1 Metodología de Evaluación para el Retardo medio

Para evaluar el retardo medio de los paquetes primero debió analizarse el proceso que se ejecuta en los Encaminadores *IntServ* e *IntServ6* (Este análisis ya fue realizado para el

primer modelo desarrollado con respecto al rendimiento del Encaminador). Posteriormente se pasó a hacer un análisis de las condiciones de evaluación y se determinaron las siguientes condiciones:

- *Tipo de Tecnología usada en el Encaminador:* Se utiliza el procesador de red IXP1200 de Intel. La justificación de esta tecnología ya fue explicada en secciones anteriores.
- *Proceso a Modelar:* Debido a que las diferencias fundamentales de la propuesta se encuentran en el proceso de clasificación, se planteó desarrollar modelos (matemáticos y de simulación) del clasificador y utilizar modelos desarrollados en otros trabajos [86] para el modelo estocástico del planificador.
- *Tipo de modelo a utilizar:* Con fines de tener fácil tratamiento matemático se decidió utilizar llegadas exponenciales y tratar de analizar la distribución del tiempo de servicio. De esta manera podría obtenerse un modelo M/G/1, o si fuese posible, un modelo M/M/1. Los modelos obtenidos permitirían entonces una comparación de las dos tecnologías de forma estadística. Aunque en principio las llegadas exponenciales no modelan de forma real el tráfico de datos en un Encaminador del núcleo de Internet, se puede considerar que bajo ciertas condiciones consideradas en [87], el tráfico en un Encaminador de una Red de Acceso podría comportarse con llegadas exponenciales de ráfagas de paquetes, de manera que si en el Encaminador se da un tratamiento por ráfagas en lugar de un tratamiento por paquetes, podrían obtenerse resultados similares a los del modelo desarrollado.

A continuación, se desarrolló un modelo matemático teórico del proceso de clasificación para las dos tecnologías (*IntServ* e *IntServ6*). Este modelo está basado en una cola M/M/1 y fue desarrollado a partir de las características del proceso de Clasificación y del procesador de red utilizado. Seguidamente, se desarrolló un modelo de simulación del Clasificador para las dos tecnologías de QoS (*IntServ* e *IntServ6*) y se analizaron los resultados. Se simuló el *Tiempo de Servicio* y el *Tiempo de Retardo* de los paquetes en el Clasificador. Para ambas tecnologías se comparó el comportamiento

del *Tiempo de Servicio* simulado del *Clasificador* con el modelo teórico con distribución exponencial para diferentes cantidades de flujos, con el objetivo de determinar si la distribución de este parámetro se ajusta a tal modelo, de manera que se pudiese utilizar un tipo de cola M/M/1. Se realizaron pruebas estadísticas y se validó esta aproximación.

Una vez verificado el modelo M/M/1 del *Clasificador*, se obtuvo una red de colas M/M/1 en *Tandem* compuesta del *Clasificador* y el *Planificador*, lo que permite determinar un *Límite Superior de Retardo Medio* en el *Encaminador*. Finalmente, los pasos anteriormente descritos serán detallados en las secciones siguientes.

4.3.2 Desarrollo del modelo matemático del Encaminador

Otra forma de evaluar el proceso de *Clasificación* fue la realización de un modelo matemático que debió ser validado estadísticamente. De esta manera se obtuvieron expresiones matemáticas que describen el proceso y permiten hacer proyecciones sobre el comportamiento del sistema sin necesidad de realizar simulaciones.

4.3.2.1 Modelo de un Encaminador de Servicios Integrados

Un *Encaminador de Servicios Integrados* está compuesto principalmente por dos partes; la primera parte es el *gestor de las reservas*, el cual está compuesto por el *Esquema de Control de Admisión (CAC)* y el *Agente de Establecimiento de las Reservas*. La segunda parte del *Encaminador* realiza el proceso de retransmisión de paquetes y está compuesta por el proceso de *Clasificación* y el proceso de *Planificación de paquetes*. Como se observa en la Figura 4.7, estas dos partes principales están relacionadas entre sí a través de la *Tabla de Reservas de Flujos*. Por tanto, cuando el módulo CAC acepta una *Nueva Reserva* (ó *Nueva Llamada*), el número de flujos en la *Tabla de Reservas de flujos* se incrementa en uno. Como resultado, cuando llega un nuevo paquete, este debe ser clasificado, es decir, el paquete debe ser buscado en la *Tabla de reserva de flujos*. A continuación, si el paquete pertenece a algún flujo deberá ser enviado al *Planificador*

del *enlace de salida* correspondiente. En este modelo, para simplificar, suponemos que el *tiempo de retardo* del *módulo de Control de Políticas* es cero.

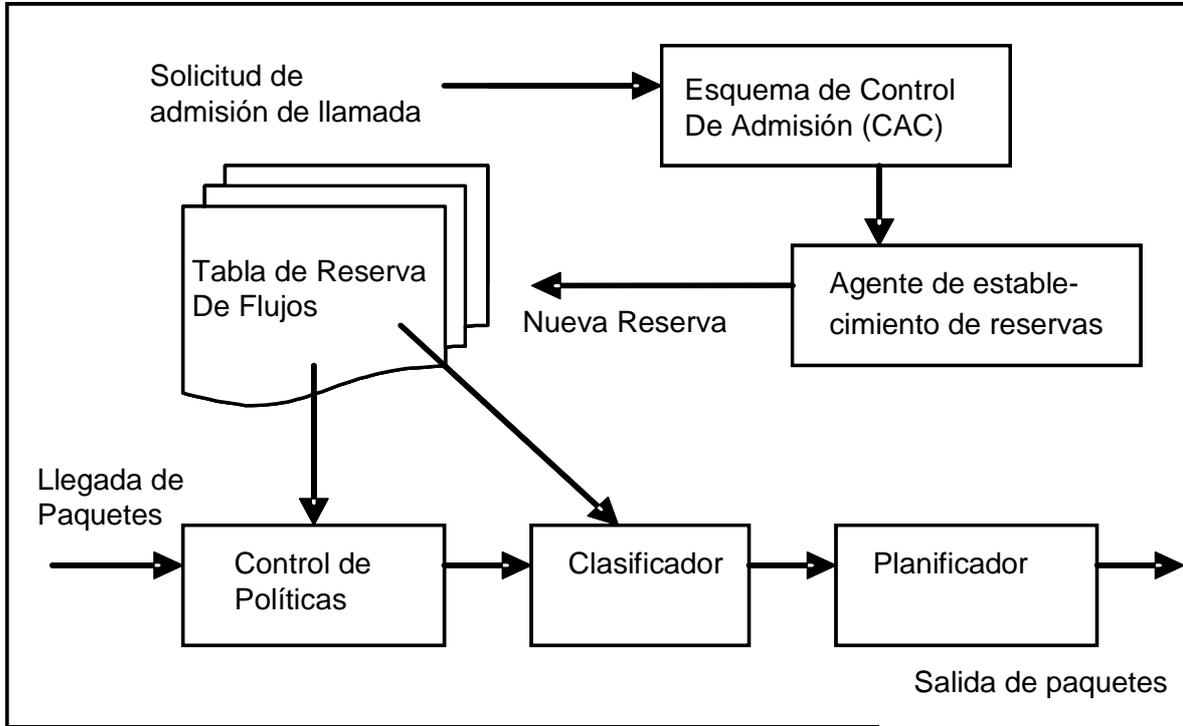


Figura 4.7 Modelo de un Encaminador de *Servicios Integrados*.

El objetivo de esta sección es obtener un modelo del retardo de los paquetes cuando estos están siendo retransmitidos por el *Encaminador*. Por tanto, sólo se tiene en cuenta el retardo del proceso de *clasificación* y de *planificación* y se obtiene una expresión para el *retardo medio* en función del *número medio de reservas*. El *número medio de reservas* es controlado a su vez por el *módulo de Control de Admisión*, por lo que en secciones posteriores se analizará el efecto del *módulo de control de Admisión* para el caso específico de *Encaminadores de Acceso Inalámbricos*.

4.3.2.2 *Modelo Estocástico del proceso de retransmisión de paquetes*

Para obtener el *retardo medio de los paquetes* es necesario obtener el modelo de colas de un *Encaminador de Servicios Integrados*. Dicho modelo se plantea a partir de los estados del procesamiento en la *retransmisión* de los paquetes. Durante esta actividad,

un *Encaminador de Servicios Integrados* tiene dos etapas para el procesamiento de los paquetes (ver Figura 4.8), una primera etapa es la de *clasificación* de los paquetes, mientras que la segunda etapa es la de *planificación* de los paquetes en cada uno de los enlaces de salida.

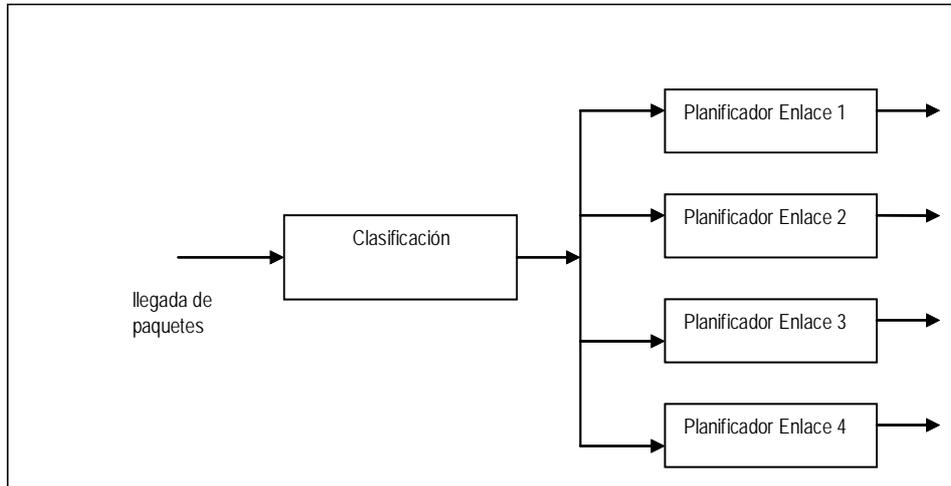


Figura 4.8 Esquema general del proceso de *retransmisión* de paquetes en un *Encaminador de Servicios Integrados*

Como resultado, se debe obtener un *modelo de colas* para la *clasificación* y otro para la *planificación*. Sin embargo, el *planificador* también es a su vez, uno de los elementos más complejos de analizar, en especial el tipo de planificador utilizado por los *Encaminadores de Servicios Integrados*, conocido como *WFQ* (*Weighted Fair Queuing*) o también *GPS* (*Generalized Processor Sharing*). Inicialmente, Parekh y Gallager en [36, 88] han hecho estudios acerca de cuál es el *peor retardo*, en un planificador *WFQ* y en una red de planificadores en cascada respectivamente. Desafortunadamente, este parámetro nos da una idea del peor caso, pero no permite dar una idea de los valores medios de *retardo* y *ancho de banda*, los cuales son muy importantes a la hora de especificar los servicios de una red de *Servicios Integrados*. Es así, como Yaron y Sidi en [89] plantean un método para caracterizar de alguna manera un sistema tan complejo. La teoría desarrollada por estos autores utiliza ráfagas de paquetes limitadas exponencialmente y se conoce como *EBB* (*EBB- Exponentially Bounded Burstiness*). Esta teoría [90] plantea que cuando las entradas a un elemento de red aislado están todas limitadas, sus resultados serán salidas limitadas, y se asegura que

los *retardos* y las *longitudes de las colas* en este elemento tengan distribuciones con decaimiento exponencial. Adicionalmente, en algunos tipos de redes, las entradas limitadas resultan en salidas limitadas y puede demostrarse que los procesos naturales de tráfico satisfacen tales límites. En consecuencia, este método permite el análisis de diferentes casos anteriormente intratables. En esta teoría, se proveen suficientes condiciones para la estabilidad de tales redes, y se derivan *límites superiores* para los parámetros de interés del desempeño de las redes.

Por otro lado, Zhang y Towsley en [91] utilizan el modelo de procesos *EBB* introducido por Yaron para obtener resultados que extienden los resultados determinísticos de Parekh y Gallager mencionados anteriormente. Es así, como Zhang y Towsley presentan *límites estadísticos* en las distribuciones de la *longitud de la cola* y el *retardo de paquetes* en cola para el caso de planificadores *GPS*.

El modelo que utilizaremos para la etapa de planificación es el obtenido en [86], el cual obtiene un modelo estocástico y una expresión para el *límite superior del retardo medio* en un planificador *PGPS* (*Packetized Generalized Processor Sharing*).

Por otro lado, se estudiará el *Retardo Medio* de los paquetes en el proceso de *Clasificación*. Para ello, se partirá del *Diagrama de Transición de Estados (DTE)* del *clasificador* en dos casos diferentes: *IntServ* e *IntServ6*. Luego se obtendrán las expresiones del *Tiempo Medio de Servicio*, para finalmente obtener la expresión del *Retardo Medio de los Paquetes* en el *Clasificador* para ambas tecnologías.

Una vez estudiados los modelos estocásticos para los procesos de *clasificación* y de *planificación*, obtendremos el modelo completo del *Encaminador* para un flujo específico (ver Figura 4.9) aprovechando que hemos obtenido un modelo de cola *M/M/1* para cada uno de ellos (con base en las condiciones y resultados descritos en las secciones 4.3.2.3 y 4.3.3) y entonces podemos aplicar un modelo de *Colas en Tandem* [92]. Así, se obtiene una expresión para calcular el *Retardo Medio Extremo a Extremo* de los paquetes en una red de colas *M/M/1* en *Tandem* bajo los supuestos del *Principio de Independencia de Kleinrock* (1976) y el *Teorema de Burke*. Para mayores detalles se puede consultar [92]. Se puede asumir independencia de los dos procesos pues el

Clasificador sólo requiere procesar la cabecera de los paquetes (por tanto, el *Tiempo de Servicio* no depende de la longitud de los mismos), mientras que el *Tiempo de Servicio* del *Planificador* depende de la *Longitud de los Paquetes*. Así, la expresión obtenida para el retardo medio extremo a extremo será de la forma,

Ecuación 4.7:

$$E(T) = \sum_{i=1}^k \frac{1}{\mu_i - \lambda_i}$$

Donde μ_i y λ_i son la *Tasa de Servicio* y la *Tasa de Llegadas* de la cola i respectivamente. Además, k es el número de enlaces en el circuito virtual y por tanto el número de colas en serie.

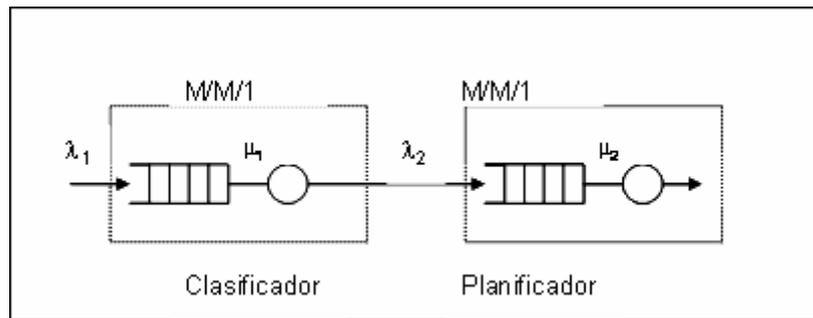


Figura 4.9 Modelo Estocástico del *Encaminador IntServ6*

4.3.2.3 Justificación del uso de llegadas exponenciales para los modelos propuestos

Los modelos planteados en esta tesis utilizan las teorías de Roberts [87] sobre transferencia de documentos (ficheros, páginas web). La transferencia de estos documentos produce ráfagas de paquetes de un mismo flujo en *Encaminadores de Redes de Acceso*, mientras que ya en los *Encaminadores del núcleo de la red*, estas ráfagas de paquetes se van rompiendo. Debido a que *IntServ6* es una tecnología fundamentalmente diseñada para *Redes de Acceso*, es posible desarrollar modelos del tratamiento por *Ráfagas de Paquetes* en el *Clasificador* en lugar del procesamiento de paquetes individuales, lo que simplifica los modelos obtenidos para evaluar esta tecnología. Por esta razón, a continuación se hace un breve recuento de los conceptos introducidos por Roberts en [87].

➤ *Teoría de Roberts sobre el Modelado de Fuentes:*

Las aplicaciones que se espera produzcan la mayoría del tráfico en la futura *Internet* multi-servicio, pueden ser clasificadas como de tipo *Streaming* (de flujo constante) o de tipo *Elástico*, de acuerdo con la naturaleza de los flujos que ellas produzcan. Los *Flujos* de tipo *Streaming* son producidos por aplicaciones de audio y video tanto para comunicaciones de tiempo real como para la reproducción de secuencias almacenadas (*Playback*). Su *Calidad de Servicio* está determinada por el grado de transparencia del camino de comunicación con respecto a la integridad de la señal inicial. Los *Flujos Elásticos*, por otro lado, resultan de la transferencia de documentos digitales (Páginas *Web*, *MP3*, *e-mails*, etc). Además, su tasa de transmisión es adaptable y depende de los niveles de tráfico del momento.

En [87] se consideran *Flujos Elásticos* exclusivamente y, con el objeto de simplificar, se ignora el impacto del tráfico tipo *Streaming* sobre la capacidad disponible. Cada *Flujo Elástico* corresponde a la transferencia de un único documento, caracterizado simplemente como un volumen de datos en bits. Un *Flujo* también puede ser definido como una secuencia de paquetes que pertenecen a una instancia de una aplicación. Así, un *Flujo* podría corresponder a una conexión *TCP* establecida para la transferencia de un elemento de un documento *Web*; incluso una página entera de este puede identificarse como una entidad única. Tal *Flujo* no se materializa en una red de *Datagramas*, sin embargo esto conlleva a que se modele el tráfico en términos de *Flujos* debido a que la *QoS*, medida en términos del tiempo de respuesta, es experimentada a este nivel (nivel de *Flujo*). Adicionalmente, suele ocurrir que todos los paquetes de un *Flujo* dado sigan el mismo camino en la red, situación en la cual el tiempo de respuesta está ampliamente determinado por el número y características de otros flujos con los cuales se comparten los recursos del camino.

Los estudios descritos en [87] adoptan un modelo de fluidos para el tráfico donde la *Tasa* de cada *Flujo* está siempre bien definida y varía dependiendo del *Número de Flujos* que comparten el ancho de banda del enlace. Allí se ignora la imprecisión que es introducida por los protocolos de nivel de paquete (p.ej. el algoritmo *Slow-Start* de

TCP) en la distribución de recursos. Esto se hace con el fin de obtener resultados simples acerca de la *QoS* a alto nivel. En consecuencia, un *Flujo* está caracterizado simplemente por un *Tiempo de Llegada* y un *Volumen de Datos* a ser transmitido en un camino de una red. Este podría ser caracterizado adicionalmente por parámetros tales como el *RTT (Round Trip Time)* y otros factores externos que afectan el *Ancho de Banda* que el *Flujo* obtiene sobre un enlace compartido.

Una razón clave para elegir modelar *Flujos* en lugar de paquetes es que la *Calidad de Servicio* es experimentada por los usuarios a este nivel. Generalmente, los usuarios de aplicaciones elásticas no son sensibles al *Retardo Extremo a Extremo* de cada paquete, sino al tiempo necesario para transferir un documento entero, que a su vez es igual al *Tiempo de Respuesta* del flujo asociado. Como se mencionó antes, esta *QoS* percibida depende de forma crítica del *Número de Flujos* que comparten los mismos recursos. Este *Número de Flujos* varía en la medida que unos inician y otros terminan. El objetivo de los modelos estudiados en [87] es estudiar la dinámica de estos flujos.

Los flujos generalmente no se dan aisladamente y son generados dentro de sesiones. Una *Sesión* podría ser definida como una serie de *Flujos* y *Tiempos de Inactividad*. Las propiedades estadísticas de una *Sesión*, incluyendo las distribuciones del *Tamaño de los Flujos* y las correlaciones entre *Flujos* sucesivos y *Tiempos de Inactividad* pueden ser complejas y claramente dependen de la aplicación subyacente. Sin embargo, las propiedades estadísticas son independientes de una sesión a otra. Este supuesto de independencia lleva naturalmente a un proceso de *Llegadas de Sesiones* con una distribución de *Poisson* cuando el número de usuarios es grande y ningún usuario genera una porción significativa del tráfico total.

Los procesos de *Llegadas de Flujos*, por otro lado, tienden a tener un comportamiento a ráfagas y se ha demostrado que tienden a un comportamiento auto-similar en ciertos casos [93, 94]. Una posible explicación para este comportamiento es que el número de flujos por sesión tiene una distribución de cola pesada [95]. Sin embargo, podría ser apropiado en algunas circunstancias suponer *Flujos* que lleguen de acuerdo a un proceso de *Poisson*. Este podría ser el caso, por ejemplo, cuando los *Flujos* corresponden a un gran número de *Sesiones* y el espaciamiento de los *Flujos* dentro de una sesión es

grande comparado con el *Intervalo entre Flujos*. Además, los resultados derivados para *Llegadas de Flujos de Poisson* son aplicables bajo el supuesto relajado de *Llegadas de Sesiones* de tipo *Poisson*. La *Tasa de Llegadas de Flujos* se denota como λ . Situaciones como éstas suceden comúnmente en Redes de Acceso en las que los tiempos de inactividad de los usuarios y los tiempos de descarga de documentos son considerables cuando se están realizando actividades de navegación por la Web, descargas de ficheros e interacción con el correo electrónico.

Las medidas del *Tamaño de los Documentos* tales como páginas Web y ficheros FTP muestran que su distribución es la de una cola pesada [94, 96]. Una consecuencia de esto es que la gran mayoría de *Flujos* son muy pequeños mientras que la mayoría del tráfico está contenido en *Flujos* grandes. La distribución precisa depende claramente del tipo de documento considerado. En los modelos desarrollados en [87], se usan diferentes distribuciones de *Tamaño de Documentos* para ilustrar el impacto de las características de este tráfico en el rendimiento o para facilitar la tratabilidad cuando las soluciones están disponibles sólo para sistemas Markovianos. El *Tamaño del Documento* se representa por la variable aleatoria σ . Un ajuste razonable a la forma de cola pesada en la práctica está dado por la distribución de *Pareto*:

Ecuación 4.8:

$$P_r[\sigma > x] = \left(\frac{b}{x}\right)^a \quad \text{para } x > b$$

Donde $a > 1$ es un parámetro fijo y b es el tamaño mínimo de documento y es igual a: $(a-1)E[\sigma]/a$. Nótese que la varianza asociada es infinita tan pronto como $a \leq 2$. En [87] se toma $a=1.4$ para aplicaciones numéricas.

➤ *Aplicabilidad a los modelos de esta tesis:*

En esta tesis, una *Reserva* o *Flujo* equivale a una *Sesión* de las teorías de Roberts, mientras que una *Ráfaga de Paquetes* equivale a un *Flujo* planteado en [87]. Por tanto, se pueden plantear *Llegadas de Reservas* con una distribución de *Poisson*, mientras que las *Ráfagas de Paquetes* tienen una *Longitud* (equivalente al *Tamaño de los Documentos*) con una distribución de *Pareto*. Además, de acuerdo con la teoría de Roberts, podría ser apropiado en algunas circunstancias suponer *Ráfagas de Paquetes*

que lleguen de acuerdo a un proceso de *Poisson*. Este podría ser el caso, por ejemplo, cuando las *Ráfagas* corresponden a un gran número de *Reservas* y el espaciamiento de las *Ráfagas* dentro de una *Sesión* es grande comparado con el intervalo entre *Ráfagas*. Además, los resultados derivados para *Llegadas de Ráfagas de Poisson* son aplicables bajo el supuesto relajado de *Llegadas de Sesiones* de tipo *Poisson*. Bajo los anteriores supuestos, y con el fin de obtener un modelo matemático más simple, es posible utilizar un modelo de llegadas exponenciales de *Ráfagas* y un procesamiento por *Ráfagas* en el módulo de *Clasificación del Encaminador*. Esto se logra realizando un tratamiento preliminar de los paquetes entrantes al *Encaminador*, de forma que se utiliza una memoria temporal y se almacenan las *Ráfagas* de paquetes, por lo que se permite al *Clasificador* hacer sólo un proceso de Clasificación por cada *Ráfaga*. El tiempo que toma este proceso no se toma en cuenta para la evaluación ya que es el mismo para ambas tecnologías y no produce una diferencia en el retardo de procesamiento. Finalmente, para simplificar la explicación, en el resto del documento se hablará de *Llegadas de Ráfagas* o *Llegadas de Paquetes* y se asumirán como expresiones equivalentes.

4.3.2.4 Tiempo medio de Servicio de los paquetes en el proceso de clasificación en IntServ6

Para obtener el *retardo medio* de los paquetes, se debe iniciar por obtener el *tiempo medio de servicio*. Este parámetro se obtiene del *Diagrama de Transición de Estados (DTE)* del proceso. El *DTE* del proceso de *clasificación de paquetes* de un *Encaminador IntServ6* se muestra en la Figura 4.10 en la que se observan 4 estados que se explican a continuación.

El estado inicial es “*Búsqueda en tabla hash y verificación de colisión*”; en este estado, el *Encaminador* toma un paquete *IPv6*, lee la *Etiqueta de Flujo* y usa esta como índice de búsqueda en la *Tabla Hash*, en la que se encuentran todos los flujos con reserva. El resultado de esta búsqueda puede ser que no se encuentra ningún flujo con esta identificación, en cuyo caso el paquete debe ser tratado como un paquete con servicio *best-effort*, por lo que el *Encaminador* pasará al estado “*Encaminamiento normal Best-Effort*”. Otro posible resultado de la búsqueda es que el flujo está en la *Tabla Hash* y

además no hay *colisión* (se verifica el campo de *colisión* en la *Tabla Hash*); en esta situación el *Encaminador* lee el campo apuntador a la *Tabla de Reservas* y pasa al estado “*Encaminamiento de paquete*”, en el cual se hace una lectura de los parámetros de *QoS* y *enlace de salida* del flujo para luego enviar este paquete al *planificador* del respectivo *enlace de salida*. Una tercera posibilidad que puede darse como resultado de la búsqueda en la *Tabla Hash* es que se encuentran flujos con esta etiqueta pero existe *colisión*. En este caso, la *colisión* deberá resolverse en el estado “*Búsqueda tabla colisiones*”, en el cual se inicia una búsqueda elemento por elemento en la *Tabla de Resolución de Colisiones* hasta encontrar el elemento que corresponde con la *Etiqueta de Flujo* y con la *Quíntupla* del paquete. Una vez encontrado dicho elemento, se obtiene de esta tabla el apuntador a la *Tabla de Reservas* y se pasa nuevamente al estado “*Encaminamiento del paquete*” que ya fue explicado.

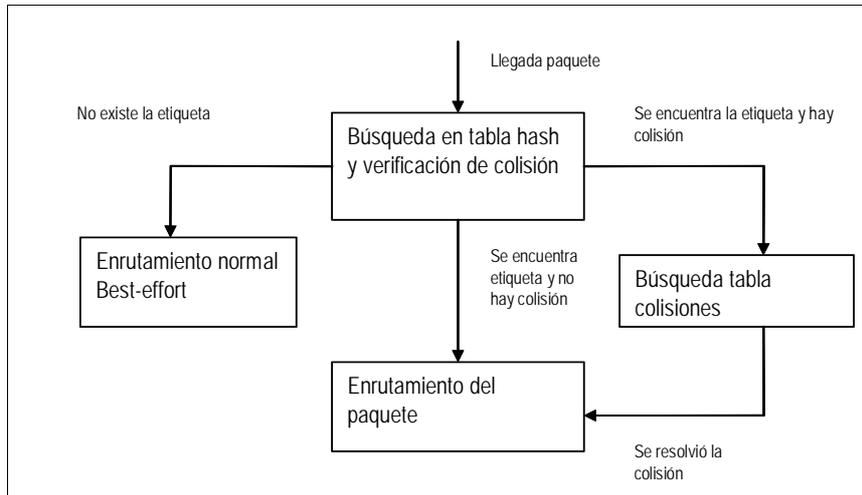


Figura 4.10 Diagrama de transición de estados del proceso de *clasificación de paquetes* de un *Encaminador IntServ6*.

Con base en el *DTE* anterior, el tiempo de servicio (x_c) del servidor que realiza la clasificación se puede calcular como:

Ecuación 4.9:

$$x_c = (t_i + t_e) (1 - C_h) + (t_i + t_e + t_x) C_h$$

y distribuyendo se obtiene :

Ecuación 4.10:

$$x_c = t_i + t_e + t_x C_h$$

La Ecuación 4.9 está dividida en 2 partes, el término $(t_i + t_e) (1-C_h)$, que es el tiempo que se tarda el proceso de *clasificación* en caso de no haber *colisión*, y $(t_i + t_e + t_x) C_h$, que es el tiempo que se tarda el proceso de *clasificación* cuando hay *colisión* con el flujo al que pertenece el paquete. Además, las diferentes variables que intervienen en la Ecuación 4.9 y la Ecuación 4.10 son:

- x_c : *tiempo de servicio* del proceso de clasificación *IntServ6*.
- t_i : tiempo que tarda el proceso de *verificación de colisión* en la *Tabla Hash*. Este tiempo es fijo, pues la *Etiqueta de Flujo* se usa como índice de búsqueda. Por tanto, sólo se hace una lectura de la tabla en que se lee el campo de colisiones.
- t_e : es el tiempo que se toma el proceso de Encaminamiento del paquete. Este tiempo es fijo y consiste en la lectura del apuntador a la *Tabla de Reservas*, seguido por la operación de lectura de la información del enlace de salida y la entrega del paquete al proceso *planificador* del enlace de salida correspondiente.
- t_x : es el tiempo que tarda el proceso de búsqueda en la *Tabla de Colisiones*. Este proceso consiste en la búsqueda de la *etiqueta* fila por fila hasta encontrar la identificación correcta (*etiqueta* y *Quíntupla*). Luego se lee el apuntador a la *Tabla de Reservas* y por último se entrega esta información al proceso de *Encaminamiento del paquete*. El tiempo t_x se puede calcular como,

Ecuación 4.11:

$$t_x = k t_{comp}$$

Donde k es el *número de intentos* hasta obtener el primer acierto, y t_{comp} es el tiempo que tarda el procesador en hacer la comparación de los datos de la *Quíntupla*.

- C_h : es la *tasa de colisiones* para m flujos, es decir, es la proporción del total de flujos activos que obtienen un *número hash* que ya ha sido asignado para al menos otro flujo [22]. Esta tasa se puede calcular como:

Ecuación 4.12:

$$C_h = 1 - \frac{N * \left(1 - \left(\frac{N-1}{N} \right)^m \right)}{m}$$

Donde N es el tamaño de la *Tabla Hash*, que para el caso de *IntServ6* es de 2^{20} ya que el la *Etiqueta de Flujo* es de 20 bits.

Para determinar la distribución del tiempo de servicio x_c , observamos que t_i y t_e son tiempos fijos, mientras que t_x es un tiempo aleatorio que está dado por la probabilidad de encontrar el primer acierto al comparar uno por uno los elementos de la tabla. Esto es el comportamiento de una *Distribución Geométrica*, ya que tal distribución se define como una secuencia de intentos independientes en que la variable aleatoria cuenta el número de intentos hasta incluir el primer éxito [92]. La ecuación de esta distribución está dada por:

Ecuación 4.13:

$$P[x = k] = (1 - p)^{k-1} p$$

Donde p es la *Probabilidad de Éxito* y $(1-p)$ la probabilidad de fracaso. Además k es el *Número de Intentos* realizado incluyendo el último. La *Probabilidad de Éxito* en la búsqueda a su vez depende del *número de flujos activos* que haya en un determinado momento y su dependencia se puede observar en la Ecuación 4.14:

Ecuación 4.14:

$$p = \frac{1}{mC_h}$$

En esta ecuación se observa que la *Probabilidad de Éxito* es el inverso de mC_h , el cual es el *Número de Flujos Colisionados*, calculado como el *Número de Flujos Activos*

multiplicado por la *proporción de flujos que colisionan*. Si se reemplaza la Ecuación 4.12 en la Ecuación 4.14 obtenemos la expresión de p en función del *Número de Flujos Activos* en la Ecuación 4.15:

Ecuación 4.15:

$$p = \frac{1}{m - \frac{1}{N^{m-1}}}$$

Donde N es conocido y constante (2^{20} combinaciones) para *IntServ6*, por lo que p depende solamente del *Número de Flujos Activos* (m).

Con base en la Ecuación 4.10, y teniendo en cuenta la distribución del *Tiempo de Servicio* de la Ecuación 4.13, podemos calcular el *Tiempo Medio de Servicio* y la varianza fácilmente. Así, el tiempo medio de servicio será,

Ecuación 4.16:

$$\bar{x}_c = E[x_c] = E[t_i + t_e + t_x C_h]$$

Ya que t_i y t_e son constantes, sus medias son los mismos valores t_i y t_e respectivamente. Para el caso de t_x , partimos de la Ecuación 4.11, donde t_{comp} es constante, pues depende de las características del procesador, mientras que la *Tasa de Colisiones* (C_h) es constante suponiendo que el *Número de Flujos activos* (m) no varía. Bajo estas condiciones, el valor medio de t_x será,

Ecuación 4.17:

$$\bar{t}_x = t_{comp} \bar{k} = t_{comp} \frac{1}{p} = t_{comp} m C_h$$

Donde el valor medio de k es el valor medio para una *Distribución Geométrica* del *número de intentos* realizados para encontrar la entrada correcta en la *Tabla de Resolución de Colisiones* expresada en la Ecuación 4.13, es decir $(1/p)$, y p se reemplazó por su equivalente en la Ecuación 4.14.

Por tanto, reemplazando la Ecuación 4.17 en la Ecuación 4.16, y teniendo en cuenta que t_i y t_e son constantes, el *Valor Medio del Tiempo de Servicio* en función del *Número de Flujos* y la *Tasa de Colisiones* será,

Ecuación 4.18:

$$\bar{x}_c = t_i + t_e + t_{comp} m C_h^2$$

Por otro lado, para calcular la varianza del *Tiempo de Servicio* debemos tener en cuenta que en la Ecuación 4.10, t_i y t_e son constantes, por lo que el calculo de la varianza para estos componentes de la ecuación es cero. Así, la varianza dependerá únicamente del componente $t_x C_h$ de esta ecuación.

Por tanto, la varianza del tiempo de servicio x_c se calcula como,

Ecuación 4.19:

$$\text{var}[x_c] = \text{var}[t_i + t_e + t_x C_h] = \text{var}[t_i] + \text{var}[t_e] + \text{var}[t_x C_h]$$

Ecuación 4.20:

$$\text{var}[x_c] = \text{var}[t_x C_h] = \text{var}[k t_{comp} C_h]$$

Como t_{comp} es constante y C_h lo suponemos constante debido a que hemos supuesto un *Número de Flujos activos* constante, y además k tiene una *Distribución Geométrica*, la Ecuación 4.20 se convierte en,

Ecuación 4.21:

$$\text{var}[x_c] = t_{comp} C_h \text{var}[k] = t_{comp} C_h \left(\frac{1}{p^2} - \frac{1}{p} \right)$$

Y reemplazando la Ecuación 4.14 en la Ecuación 4.21 obtenemos la varianza de x_c en función de la *Tasa de Colisiones* y el *Número de Flujos* en la siguiente expresión,

Ecuación 4.22:

$$\text{var}[x_c] = t_{comp} m C_h^2 (m C_h - 1)$$

4.3.2.5 *Tiempo medio de Servicio de los paquetes en el proceso de clasificación en IntServ*

El modelo obtenido para el caso de *IntServ6* sirve como base para obtener el modelo para *IntServ*. Sin embargo, existen diferencias entre *IntServ* e *IntServ6* en el proceso de *Clasificación*, las cuales se encuentran resumidas en la Tabla 4.2.

Tabla 4.2 Diferencias entre *InServ* e *IntServ6*

<i>Diferencia</i>	<i>IntServ</i>	<i>IntServ6</i>
El número hash se calcula en:	Cada <i>Encaminador</i> del camino de los paquetes	El host origen
Tamaño del número hash	32 bits (usando CRC)	20 bits (usa XOR holding)
Número de veces que se calcula el número hash	1 vez por paquete	1 vez por sesión
Tamaño de la tabla hash	Variable. Entre 256KB y 4MB	Fijo (1MB)
Tipo de búsqueda en la tabla hash	Matching	Tagging
Tamaño de la quintupla	104 bits	296 bits

Para calcular el tiempo de servicio para el clasificador *IntServ*, se parte del *DTE* del *Clasificador*, el cual cambia por el de la Figura 4.11.

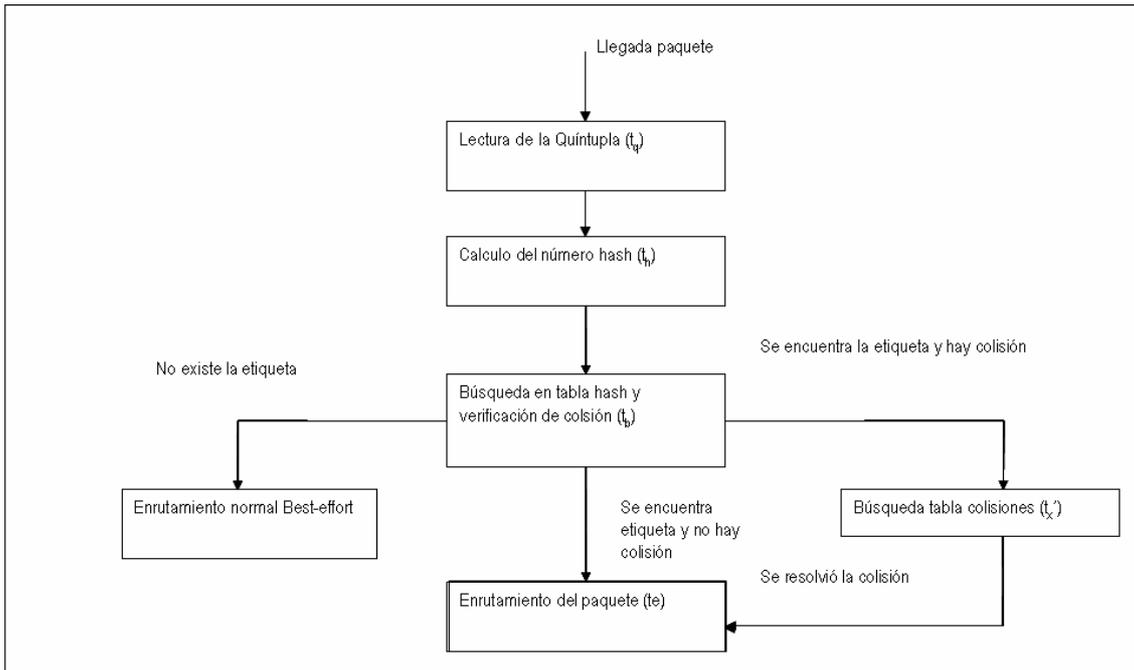


Figura 4.11 Diagrama de Transición de estados del *clasificador InServ*.

Con base en el diagrama de transición de estados del *Clasificador de IntServ*, se puede obtener el tiempo medio de servicio \bar{x}_{co} . La ecuación para este tiempo, se puede calcular como la sumatoria de los tiempos de cada proceso, multiplicada por su probabilidad de ocurrencia,

Ecuación 4.23:

$$x_{co} = t_q + t_h + (t_b + t_e)(1 - C_h) + (t_b + t'_x + t_e)C_h$$

Y reagrupando se obtiene,

Ecuación 4.24:

$$x_{co} = t_q + t_h + t_b + t_e + t'_x C_h$$

Donde t_q , t_h , t_b , t_e y t'_x son los tiempos que tarda cada uno de los procesos. Estos se pueden observar en la Figura 4.11.

Es importante aclarar que los tiempos asociados a los procesos de *lectura de la quintupla* y *cálculo del número hash*, son fijos para todo paquete y estos procesos siempre ocurren. El proceso denominado *Búsqueda en la tabla hash y verificación de*

la colisión cambia con respecto al modelo de *IntServ6*, pues en *IntServ* se utiliza el *matching* para la búsqueda de los flujos en las tablas, mientras que en *IntServ6* se utiliza el *tagging* para el mismo proceso. Por tanto, el valor medio del tiempo t_b debe ser calculado para una *distribución geométrica* para *IntServ* y no es constante como en el caso de *IntServ6*. Por su parte, el valor fijo de t'_x , que es el tiempo que tarda el proceso de *Búsqueda en la tabla de colisiones*, es ligeramente diferente al del mismo proceso en *IntServ6*, lo que se debe a que el *número hash* es de un tamaño mayor en *IntServ6* (32 bits en *IntServ* y 20 bits en *IntServ6*), y el valor de la *Quíntupla* también varía en ambas tecnologías (104 bits para *IntServ* y 296 bits para *IntServ6*).

Con estas aclaraciones, se obtiene que el valor medio del *Tiempo de Servicio* del clasificador *IntServ* será,

Ecuación 4.25:

$$\bar{x}_{co} = E[x_{co}]$$

O también,

Ecuación 4.26:

$$\bar{x}_{co} = t_q + t_h + \bar{t}_b + t_e + \bar{t}'_x C_h$$

Donde el valor medio de t_b debe calcularse como el valor medio de una *Distribución Geométrica* con probabilidad de éxito $p_b = \frac{1}{m}$, donde a su vez m es el *Número de flujos activos*. Por tanto,

Ecuación 4.27:

$$\bar{t}_b = \frac{1}{p_b} t_{comph} = m t_{comph}$$

Donde t_{comph} es el tiempo requerido para comparar una fila de la *Tabla Hash* con la *Clave Hash* del paquete. Además, el valor medio de t'_x se puede calcular de la misma forma que en el modelo de *IntServ6* (ver Ecuación 4.17), es decir,

Ecuación 4.28:

$$\bar{t}'_x = t'_{comp} m C_h$$

En la Ecuación 4.28 simplemente cambia el tiempo de comparación con respecto a la Ecuación 4.17 debido al cambio en el tamaño de los datos a leer de la tabla. Por tanto, reemplazando la Ecuación 4.27 y la Ecuación 4.28 en la Ecuación 4.26 se obtiene,

Ecuación 4.29:

$$\bar{x}_{co} = t_q + t_h + mt_{comph} + t_e + t'_{comp} mC_h^2$$

De igual forma podemos calcular la varianza del tiempo de servicio en *IntServ* como,

Ecuación 4.30:

$$\text{var}[x_{co}] = \text{var}[t_q + t_h + t_b + t_e + t'_x C_h]$$

Como t_q , t_h , t_e son constantes, su varianza es cero. Mientras que la varianza de t'_x se calcula de igual forma que en las ecuaciones comprendidas desde la Ecuación 4.20 a la Ecuación 4.22 pero cambiando el valor de t_{comp} por t'_{comp} . Así mismo, la varianza de t_b se calcula nuevamente como la varianza de una *Distribución Geométrica*, es decir,

Ecuación 4.31:

$$\text{var}[t_b] = \left(\frac{1}{p_b^2} - \frac{1}{p_b}\right)t_{comph} = (m^2 - m)t_{comph}$$

Entonces, teniendo en cuenta las observaciones realizadas para t'_x y reemplazando la Ecuación 4.31 en la Ecuación 4.30, obtenemos la expresión de la varianza del *Tiempo de Servicio* en *IntServ* así,

Ecuación 4.32:

$$\text{var}[x_c] = (t'_{comp} C_h^3 + t_{comph})m^2 - (t_{comph} + t'_{comp} C_h^2)m$$

4.3.2.6 Modelo de colas del proceso de clasificación

Para simplificar el análisis del proceso de clasificación se asumen las condiciones descritas en la sección 4.3.2.3. Es decir, se suponen aplicaciones semi-elásticas, por lo que los paquetes llegan a *Ráfagas*. Además, bajo las condiciones descritas en dicha sección, se puede asumir que las *Ráfagas* llegan con una *Distribución de Poisson*. También se supone que el *Encaminador* tiene una *Memoria Temporal* que almacena

cada *Ráfaga*, de forma que sólo se debe realizar el proceso de clasificación una vez por cada ráfaga de paquetes. Como se mencionó en la sección 4.3.2.3., estos supuestos son posibles con base en los estudios realizados por Roberts en [87], donde se plantea que tal comportamiento puede ser obtenido del tráfico de *Internet* generado a partir de conexiones *TCP* y aplicaciones tales como el *Web* y la *Transferencia de Ficheros (FTP-File Transfer Protocol)*. De esta manera, se puede obtener un modelo del proceso de clasificación como una cola *M/G/1*. Se considera una *Distribución General* para la disciplina de servicio ya que hay una *Distribución Geométrica* para el proceso de *Resolución de Colisiones* para ambas tecnologías, *IntServ* e *IntServ6*. Además, también hay una *Distribución Geométrica* en el tiempo de búsqueda en la *Tabla Hash* para el caso de *IntServ* (ya que utiliza *Matching*), más no para el caso de *IntServ6*, en donde se utiliza *Tagging*. Se considera una *Distribución Geométrica* en estos procesos debido a que se realiza una comparación con cada entrada de la tabla hasta que ocurra una coincidencia.

Como conocemos el valor medio y la varianza del *Tiempo de Servicio*, por tanto se pueden usar las expresiones de una cola de este tipo, que son:

- Tiempo de espera medio: W

Ecuación 4.33:

$$W = \frac{\lambda \bar{x}_c^2}{2(1 - \rho)}$$

Donde $\rho = \lambda \bar{x}_c$; ρ es el factor de utilización, λ es la tasa media de llegadas, \bar{x}_c es el tiempo medio de servicio y \bar{x}_c^2 es el segundo momento del *Tiempo de Servicio*.

- Número de clientes en cola: N_q

Ecuación 4.34:

$$N_q = \lambda W = \frac{\lambda^2 \bar{x}_c^2}{2(1 - \rho)}$$

- Tiempo pasado en el sistema: T

Ecuación 4.35:

$$T = \bar{x}_c + \frac{\lambda \bar{x}_c^2}{2(1-\rho)}$$

- Número de clientes en el sistema: N

Ecuación 4.36:

$$N = \rho + \frac{\lambda^2 \bar{x}_c^2}{2(1-\rho)}$$

Sin embargo, como se desean utilizar los principios de redes de colas para construir el modelo del *Encaminador*, se debería obtener un modelo $M/M/1$ para el proceso de clasificación. Por tanto, se hacen algunas suposiciones como sigue. Normalmente, cuando ocurre una colisión, el clasificador busca un *Número Hash* en la *Tabla de Colisiones*. El tiempo de búsqueda para este caso tiene una *Distribución Geométrica*. Sin embargo, si no se crean o deshacen más reservas de flujos, la posición en la *Tabla de Resolución de Colisiones* será siempre la misma para el mismo flujo. Así, el tiempo de clasificación será el mismo para los paquetes que pertenecen al mismo flujo. Pero si se considera que las reservas de flujos se crean y se deshacen de forma aleatoria, podemos asumir que la ubicación de los flujos en la *Tabla de Resolución de Colisiones* es aleatoria y exponencialmente distribuida. De igual forma se pueden hacer suposiciones similares para el caso de la *Tabla Hash* de *IntServ*. Como resultado, podemos asumir que los tiempos de servicio para ambos procesos (*Resolución de Colisiones* para *IntServ* e *IntServ6*, y para la *búsqueda Hash* en *IntServ*) tienen una distribución de *Poisson*.

Bajo estos supuestos, y también con base en la comprobación estadística de hipótesis realizada en la sección 4.3.3, podemos asumir una cola $M/M/1$ para el proceso de clasificación de una sesión específica. Para realizar este modelo, podemos asumir el mismo *Tiempo Medio de Servicio* obtenido para el modelo $M/G/1$ debido a que el

proceso de clasificación es el mismo para todos los paquetes independientemente del flujo al que pertenezcan.

A continuación se describe el modelo de la cola $M/M/1$ para el clasificador:

- Probabilidad de tener k clientes en el sistema: P_k

Ecuación 4.37:

$$P_k = (1 - \rho)\rho^k$$

- Probabilidad de tener N o más clientes

Ecuación 4.38:

$$P[N \geq n] = \rho^n$$

- Número medio de clientes en el sistema: N

Ecuación 4.39:

$$N = \frac{\rho}{1 - \rho}$$

- Tiempo medio en el sistema: T

Ecuación 4.40:

$$T = \frac{\rho}{\lambda(1 - \rho)} = \frac{1}{\mu - \lambda}$$

- Tiempo medio en cola: W

Ecuación 4.41:

$$W = \frac{\rho}{\mu - \lambda}$$

- Número medio de clientes en la cola: N_q

Ecuación 4.42:

$$N_q = \frac{\rho^2}{1 - \rho}$$

Donde la distribución del tiempo de espera es,

Ecuación 4.43:

$$F_w(t) = 1 - \rho e^{-\mu(1-\rho)t}$$

Y donde $\rho = \lambda \bar{x}_c$; ρ es el factor de utilización, λ es la tasa media de llegadas, $\bar{x}_c = \frac{1}{\mu}$ es el *tiempo medio de servicio* obtenido para cada tipo de clasificador a evaluar (*IntServ* o *IntServ6*).

4.3.2.7 Modelo del planificador

El modelo utilizado para el planificador es el obtenido en [86], que consiste en un modelo estocástico de un planificador *PGPS* para un flujo específico. En este modelo se obtuvieron expresiones para el *tiempo de servicio* de los paquetes de la sesión o flujo i (x_i), su *Valor Medio*, su *Varianza* y el *Límite Superior del Retardo Promedio* de los paquetes. Estas expresiones fueron obtenidas para una *Distribución General* de la *Longitud del Paquete* (L_i), y por tanto para una cola *M/G/1*, y luego se obtuvieron las expresiones para los casos particulares de *Distribución Exponencial* de L_i (cola *M/M/1*) y también para las distribuciones *Uniforme* y de *Pareto*. El interés del presente estudio se centra en las expresiones para una cola *M/M/1*.

Las expresiones para la distribución general (cola *M/G/1*) son:

- *Tiempo de Servicio* de los paquetes de la sesión i :

Ecuación 4.44:

$$x_i = \frac{L_i}{w_i C}$$

Donde L_i es la *Longitud del Paquete* de la sesión i , w_i es el *Peso del Servicio* para la sesión i y C es la *Capacidad Total de Salida del Servidor* (dada en bits/s).

- *Tiempo Medio de Servicio* de los paquetes de la sesión i :

Ecuación 4.45:

$$\bar{x}_i = E[x_i] = \frac{\bar{L}_i}{w_i C}$$

- Segundo momento del *Tiempo de Servicio* de los paquetes de la sesión i :

Ecuación 4.46:

$$\bar{x}_i^2 = E[x_i^2]$$

- *Límite Superior del Retardo Medio* para los paquetes de la sesión i :

Ecuación 4.47:

$$\bar{d}_i \leq \bar{x}_i + \frac{\lambda_i \bar{x}_i^2}{2(1 - \lambda_i \bar{x}_i)}$$

De igual forma, las expresiones obtenidas para la distribución exponencial (cola $M/M/1$) son:

- *Tiempo de Servicio* de los paquetes de la sesión i :

Ecuación 4.48:

$$x_i = \frac{L_i}{w_i C}$$

- *Tiempo Medio de Servicio* de los paquetes de la sesión i :

Ecuación 4.49:

$$\bar{x}_i = E[x_i] = \frac{\bar{L}_i}{w_i C}$$

- Segundo momento del *Tiempo de Servicio* de los paquetes de la sesión i :

Ecuación 4.50:

$$\bar{x}_i^2 = E[x_i^2] = 2 \left[\frac{\bar{L}_i}{w_i C} \right]^2$$

- *Límite Superior del Retardo Medio* para los paquetes de la sesión i :

Ecuación 4.51:

$$\bar{d}_i \leq \frac{\bar{L}_i}{w_i C - \lambda_i \bar{L}_i}$$

4.3.2.8 *Modelo Completo del Encaminador*

Como se mencionó anteriormente, se pueden aplicar los principios de redes de colas $M/M/1$ en *Tandem* bajo los supuestos del *Principio de Independencia de Kleinrock* y el *Teorema de Burke* [92]. Estos podemos aplicarlos y utilizar el modelo de cola $M/M/1$ del planificador *GPS* en el que se asume que la *Longitud de los Paquetes* tiene una *Distribución Exponencial* ya que esto no riñe con el modelo $M/M/1$ del clasificador pues en este modelo la *Distribución Exponencial* de la disciplina de servicio es independiente de la distribución de la *Longitud de los Paquetes*, por lo que asumir una *Distribución Exponencial* en la distribución de los mismos no afecta la distribución de la disciplina de servicio del clasificador. Sin embargo, la

Ecuación 4.7 no podemos aplicarla tal como está a nuestro modelo pues el modelo del planificador sólo permite hallar el *Límite Superior del Retardo Medio* en la cola, así que lo que podemos obtener para el *Encaminador* es un *Límite Superior del Retardo Medio* de los paquetes.

Para obtener este límite partimos de la Ecuación 4.51, que es el límite superior para el retardo medio de los paquetes en el planificador, y le agregamos el retardo medio de los paquetes en el clasificador siguiendo la regla dictada por la

Ecuación 4.7. Así obtenemos que la media del retardo del *Encaminador IntServ6* (d_r) es,

Ecuación 4.52:

$$\bar{d}_r = \bar{d}_i + \bar{d}_c$$

Donde d_i es el retardo del planificador y d_c es el retardo para el clasificador. Por tanto, si combinamos la

Ecuación 4.7, la Ecuación 4.51 y la Ecuación 4.52 tendremos,

Ecuación 4.53:

$$\bar{d}_i + \bar{d}_c \leq \frac{\bar{L}_i}{w_i C - \frac{\bar{L}_i}{\bar{x}_c}} + \bar{d}_c$$

Donde λ_i fue reemplazada en la Ecuación 4.51 por $\lambda_i = \frac{1}{\bar{x}_c}$ ya que, cuando el

clasificador está en saturación, la tasa de entrada al planificador (λ_i) es igual a la tasa de salida del clasificador (μ_1).

Además, el valor de la media de d_c debe ser calculado mediante la Ecuación 4.40 pero con base en tasas dadas en *bits/s* y no en *paquetes/s* como se expresa en esta ecuación por lo que para realizar el cambio de unidades se multiplican las tasas de llegada y salida dadas en *paquetes/s* por la media de la longitud de los paquetes en bits (L_i). Así, la Ecuación 4.53 se convierte en,

Ecuación 4.54:

$$\bar{d}_i + \bar{d}_c \leq \frac{\bar{L}_i}{w_i C - \frac{\bar{L}_i}{\bar{x}_c}} + \frac{\bar{L}_i}{\frac{\bar{L}_i}{\bar{x}_c} - \lambda_1 \bar{L}_i}$$

O también,

Ecuación 4.55:

$$\bar{d}_r \leq \frac{\bar{L}_i}{w_i C - \frac{\bar{L}_i}{\bar{x}_c}} + \frac{\bar{L}_i}{\frac{\bar{L}_i}{\bar{x}_c} - \lambda_1 \bar{L}_i}$$

La Ecuación 4.55 es entonces el *Límite Superior del Retardo Medio* para un flujo i en un *Encaminador IntServ6*. Para el caso de *IntServ* debe cambiarse x_c por x_{co} en esta expresión.

4.3.3 Simulación del proceso real del Clasificador

Para evaluar el proceso de *Clasificación* en las dos tecnologías a comparar (*IntServ* e *IntServ6*), fue necesario desarrollar un *Modelo de Simulación* que permitiese observar el comportamiento del *Tiempo de Servicio* y del *Tiempo de Retardo*. A continuación se describirá el modelo realizado y los resultados obtenidos.

4.3.3.1 Descripción del Modelo de Simulación

El *Modelo de Simulación* se observa en la Figura 4.12. Este consta de varios procesos que se ejecutan en la siguiente secuencia:

- Se genera una *Tabla Hash* con *Claves Hash* aleatorias.
- Luego, se realiza un análisis de las *Colisiones* ocurridas y se genera una *Tabla de Resolución de Colisiones*. Además, se genera una nueva *Tabla Hash* sin repeticiones de *Claves Hash* y con los indicadores de *Colisiones* en las claves que se hayan repetido.
- Se generan llegadas de paquetes con una función aleatoria. La distribución de esta función generadora puede ser de cualquier tipo. Sin embargo, con base en lo explicado en la sección anterior, y con el objetivo de obtener un sistema M/M/1, se generan llegadas de paquetes con una función exponencial cuya tasa dependerá del *Número de Flujos* a simular y de la cantidad de paquetes por segundo que genera cada *Flujo*. Para las simulaciones que se realizaron se tuvo una tasa de 128 paquetes/s para cada uno de los *Flujos*.
- Se ejecuta el proceso de *Clasificación*, el cual se alimenta de las llegadas de paquetes, y para cada una de ellas genera una *Clave Hash* aleatoria que se utiliza para identificar el *Flujo* al que pertenece el paquete. Este proceso calcula los

Tiempos de Servicio y los *Tiempos de Retardo* de cada paquete para cada una de las tecnologías de *QoS*. El cálculo de los tiempos se hace teniendo en cuenta los tiempos de lectura de memoria *RAM* estática y dinámica para cada operación realizada, siguiendo los mismos principios utilizados para el modelo del rendimiento del *Encaminador* en la sección 4.2.

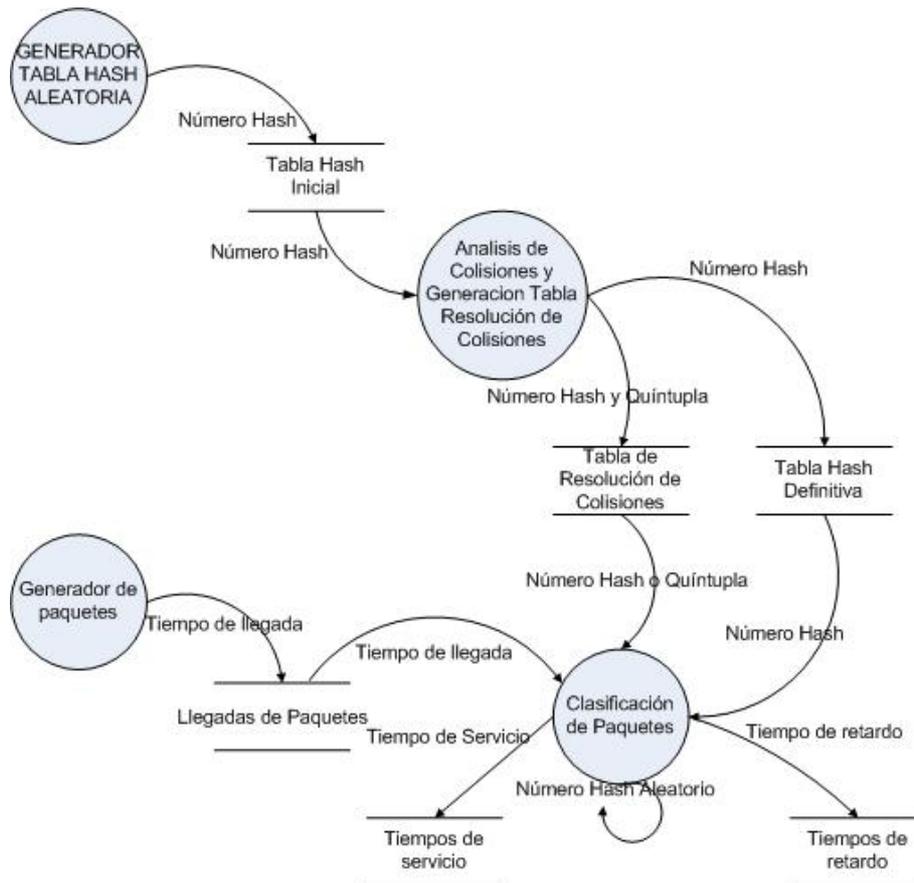


Figura 4.12 Procesos de la simulación del *Clasificador*.

Las simulaciones se llevaron a cabo para un solo *Encaminador* y fueron desarrolladas para los parámetros de un procesador de red *intel IXP1200* que están dados en la Tabla 4.3. El tamaño de la *Tabla Hash* es de 2^{20} . Este valor se asume para comparar ambas tecnologías debido a que es el tamaño de la *Tabla Hash* de *IntServ6*. Además, se usó la cabecera del protocolo *IPv6*.

Tabla 4.3 Parámetros de Simulación del procesador de red

Parámetros de Simulación del procesador de red IXP 1200	
<i>Parametro</i>	<i>Valor</i>
L_{SDRAM} (Tiempo de lectura de <i>SDRAM</i>)	160 ns
L_{SRAM} (Tiempo de lectura de <i>SRAM</i>)	80 ns
Tamaño de la palabra del procesador	32 bits
<i>Tiempos de procesamiento</i>	
$t_i = 1L_{SRAM}$;(tiempo de verificación de la <i>Tabla Hash- IntServ, IntServ6</i>)	80ns
$t_e = 2L_{SRAM} + 32L_{SDRAM}$;(tiempo de Encaminamiento del paquete, es el mismo para <i>IntServ</i> and <i>IntServ6</i>)	5280ns
$t_{comp} = 10 * L_{SRAM}$ (Tiempo de comparación de la <i>Quíntupla</i> para la resolución de colisiones)	800ns
$t_q = 10 * L_{SRAM}$ (Tiempo de lectura de la <i>Quíntupla</i> para la búsqueda en la <i>Tabla Hash</i>)	800ns

4.3.3.2 Simulaciones y Resultados Obtenidos

Como se mencionó anteriormente, los parámetros simulados fueron el *Tiempo de Servicio del Clasificador* y el *Retardo* de los paquetes. Los resultados de estas simulaciones se describen a continuación.

4.3.3.2.1 Simulaciones del tiempo de servicio

Se realizaron dos tipos de simulaciones para el *Tiempo de Servicio*. El primer tipo de simulaciones buscaba comprobar si el *Tiempo de Servicio* del *Clasificador* podría ser modelado mediante una distribución exponencial y así verificar el modelo M/M/1 teórico obtenido, por lo que debió realizarse una prueba de hipótesis. El segundo tipo de simulaciones consistió en observar el comportamiento del *Tiempo Medio de Servicio* al aumentar el *Número de Flujos*.

➤ *Prueba de Hipótesis acerca del tiempo de servicio*

Con el fin de determinar si podría utilizarse una distribución exponencial para el *Tiempo Medio de Servicio* y así validar el uso de colas M/M/1 en el modelo matemático desarrollado, fue necesario realizar el procedimiento conocido como *Prueba de Hipótesis*. En este procedimiento se tienen muestras de un proceso y estas son contrastadas con una función generadora de números aleatorios que tenga la distribución que se quiere comprobar. Existen diversos métodos para realizar estas pruebas [97] tales como la prueba *Chi-Cuadrado*, la prueba de *Kolmogorov-Smirnov* y la prueba conocida como el *Método de los Momentos* [98]. Inicialmente se intentó realizar la prueba *Chi-Cuadrado*, sin embargo, los resultados obtenidos con esta prueba no fueron buenos debido a que es una prueba muy exigente y las dos distribuciones a comparar tienen que ser casi idénticas para que arroje un buen resultado. Por tanto, se optó por una prueba que es menos exigente en el comportamiento de las dos funciones pero que a la vez permite determinar que dos distribuciones tienen un comportamiento cercano. Esta prueba se conoce como el *Método de los Momentos* y su principio consiste en comparar el primer y segundo momento de las dos distribuciones. Si estos coinciden, se puede decir que tienen un comportamiento muy cercano.

La prueba de hipótesis debió realizarse para el *Tiempo de Servicio* del *Clasificador* en diferentes situaciones que se diferencian entre sí por el *Número de Flujos* establecidos en el *Encaminador*. Los resultados de estas pruebas se encuentran en la Tabla 4.4 y la Tabla 4.5. En ellas se observa que se realizó la prueba para cantidades de flujos que varían desde 11 hasta 1200. En estas tablas se detalla el porcentaje de error para cada momento y para cada cantidad de flujos. Como puede observarse, se obtienen resultados aceptables (error $\leq 25\%$ para el segundo momento, manteniendo un error $<1\%$ en el primer momento) para el rango comprendido entre los 70 flujos y los 1200 flujos. Por tanto, se puede asumir un comportamiento con una distribución exponencial para el *Tiempo de Servicio* en estos rangos y en consecuencia, el modelo M/M/1 teórico obtenido es válido para estas condiciones.

Tabla 4.4 Método de los momentos para *IntServ*

Número de Flujos	Primer momento			Segundo momento		
	Fun.gen.exp	Valor Simulado	%Error	Fun.gen.exp	Valor Simulado	%Error
11	1,1861E-05	1,1922E-05	0,51	2,4436E-10	1,4226E-10	41,78
20	1,2247E-05	1,2300E-05	0,43	2,6052E-10	1,5172E-10	41,76
30	1,3170E-05	1,3205E-05	0,27	3,0128E-10	1,7757E-10	41,06
70	2,0121E-05	2,0200E-05	0,39	7,0318E-10	5,2663E-10	25,11
100	2,8013E-05	2,8157E-05	0,51	1,3630E-09	1,0281E-09	24,57
150	4,5584E-05	4,5800E-05	0,47	3,6091E-09	3,0672E-09	15,02
200	6,2660E-05	6,3000E-05	0,54	6,8195E-09	6,5160E-09	4,45
250	9,8078E-05	9,8600E-05	0,53	1,6708E-08	1,3954E-08	16,48
400	1,51E-04	1,52E-04	0,51	3,95E-08	3,40E-08	13,93
700	3,24E-04	3,26E-04	0,51	1,83E-07	1,45E-07	20,52
1200	4,91E-04	4,93E-04	0,51	4,19E-07	3,17E-07	24,25

Tabla 4.5 Método de los momentos para *IntServ6*

Número de Flujos	Primer momento			Segundo momento		
	Fun.gen.exp	Valor Simulado	%Error	Fun.gen.exp	Valor Simulado	%Error
11	5,46E-06	5,49E-06	0,51	5,1780E-11	3,0255E-11	41,57
20	5,3963E-06	5,4240E-06	0,51	5,0579E-11	2,9492E-11	41,69
30	5,9375E-06	5,9680E-06	0,51	6,1234E-11	3,8243E-11	37,55
70	1,2146E-05	1,2208E-05	0,51	2,5622E-10	2,6544E-10	3,60
100	1,9691E-05	1,9792E-05	0,51	6,7346E-10	6,3974E-10	5,01
150	3,6294E-05	3,6480E-05	0,51	2,2879E-09	2,3210E-09	1,45
200	5,2228E-05	5,2496E-05	0,51	4,7379E-09	5,3235E-09	12,36
250	8,7758E-05	8,8208E-05	0,51	1,3377E-08	1,1975E-08	10,48
400	1,41E-04	1,42E-04	0,51	3,47E-08	3,08E-08	11,03
700	3,14E-04	3,16E-04	0,51	1,72E-07	1,38E-07	19,74
1200	4,81E-04	4,84E-04	0,51	4,02E-07	3,06E-07	23,83

Como un complemento a la prueba de hipótesis realizada, se comprobó que los resultados del tiempo de servicio estuvieran dentro de un intervalo de confianza. Se consideró que la confianza fuese del 95% y se obtuvieron los resultados que se grafican en la Figura 4.13 y la Figura 4.14. En ellas se observa que el valor del tiempo medio de

clasificación obtenido con el modelo se encuentra dentro de los intervalos de confianza para los diferentes valores, lo que refuerza la comprobación de la hipótesis inicial de una distribución exponencial para dicho tiempo.

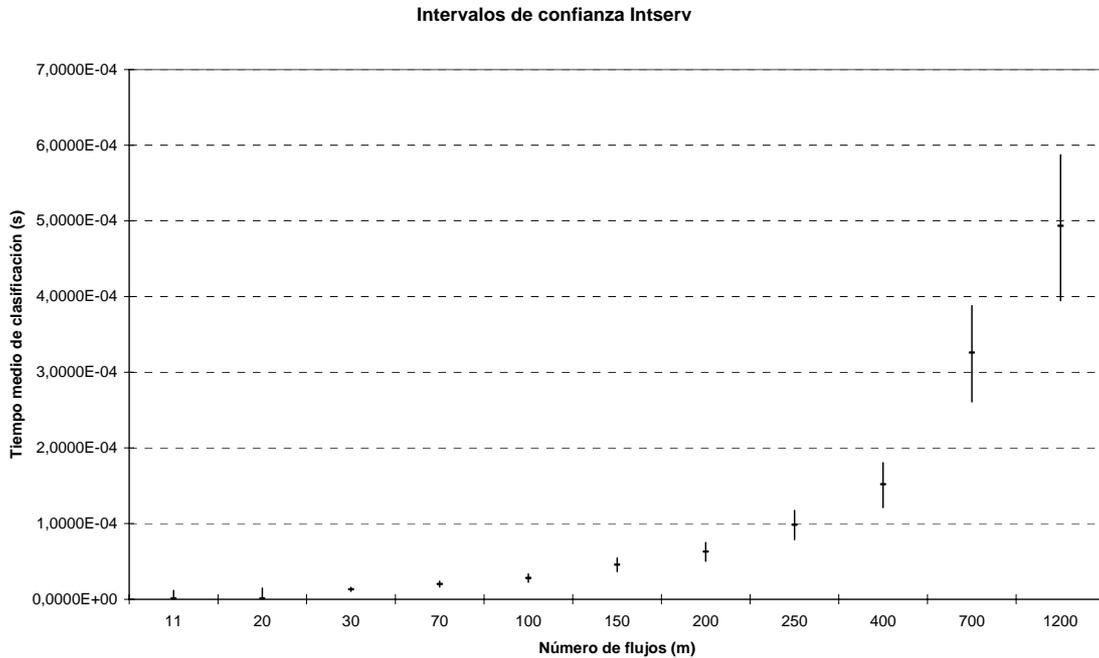


Figura 4.13 Intervalos de Confianza para *IntServ*.

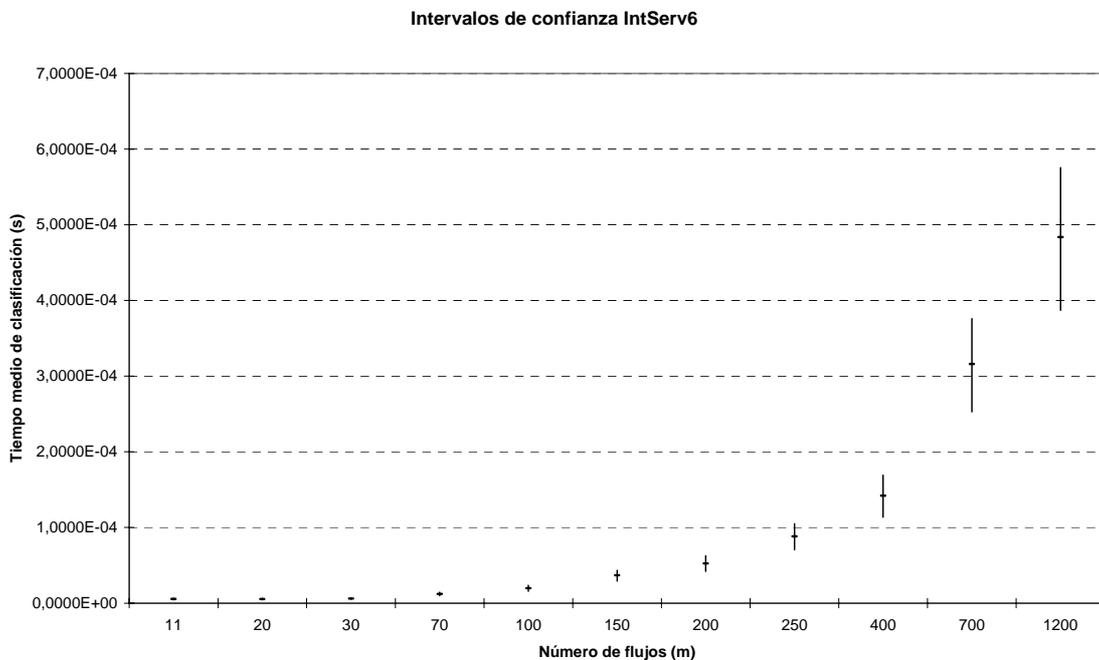


Figura 4.14 Intervalos de Confianza para *IntServ6*.

➤ *Comportamiento del tiempo medio con el número de flujos*

Otro tipo de simulaciones realizadas buscaron analizar el comportamiento del *Tiempo Medio de Clasificación* con respecto al *Número de Flujos* con reservas en el *Encaminador*. Los resultados de esta prueba se encuentran en la Figura 4.15 para *IntServ* y en la Figura 4.16 para *IntServ6*. En ellas se observa que el *Tiempo Medio de Clasificación* aumenta al incrementar el *Número de Flujos*. Para el caso de *IntServ*, esto se debe al crecimiento del número de lecturas de la *Tabla Hash* y también a que el *Número de Colisiones* aumenta. Por otro lado, para el caso de *IntServ6*, este fenómeno se debe exclusivamente al aumento del *Número de Colisiones*, ya que la *Tabla Hash* sólo es leída una vez en cada proceso de *Clasificación*.

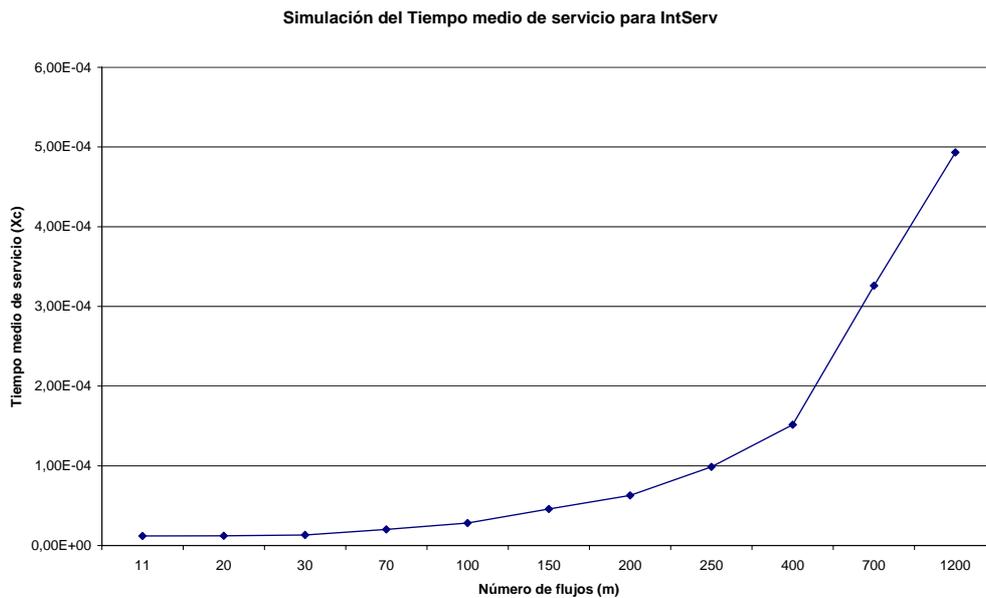


Figura 4.15 Simulación del *Tiempo Medio de Servicio* de *IntServ* respecto al *Número de Flujos* en el *Encaminador*.

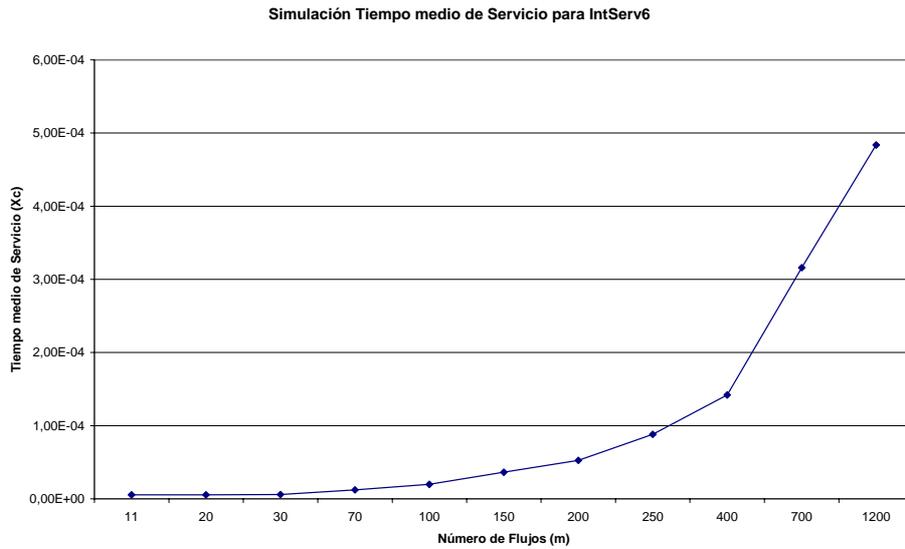


Figura 4.16 Simulación del *Tiempo Medio de Servicio* de *IntServ6* respecto al *Número de Flujos* en el *Encaminador*.

Por último, en la Figura 4.17 se observa la comparación de los tiempos de clasificación de las dos tecnologías. Se observa que el tiempo medio de clasificación para *IntServ6* es más pequeño que el mismo parámetro para *IntServ*. Esto se debe a que *IntServ6* no debe calcular la *Clave Hash* ni tampoco hacer búsquedas en la *Tabla Hash*, pues usa la *Etiqueta de Flujo* como un apuntador a una posición de esta tabla.

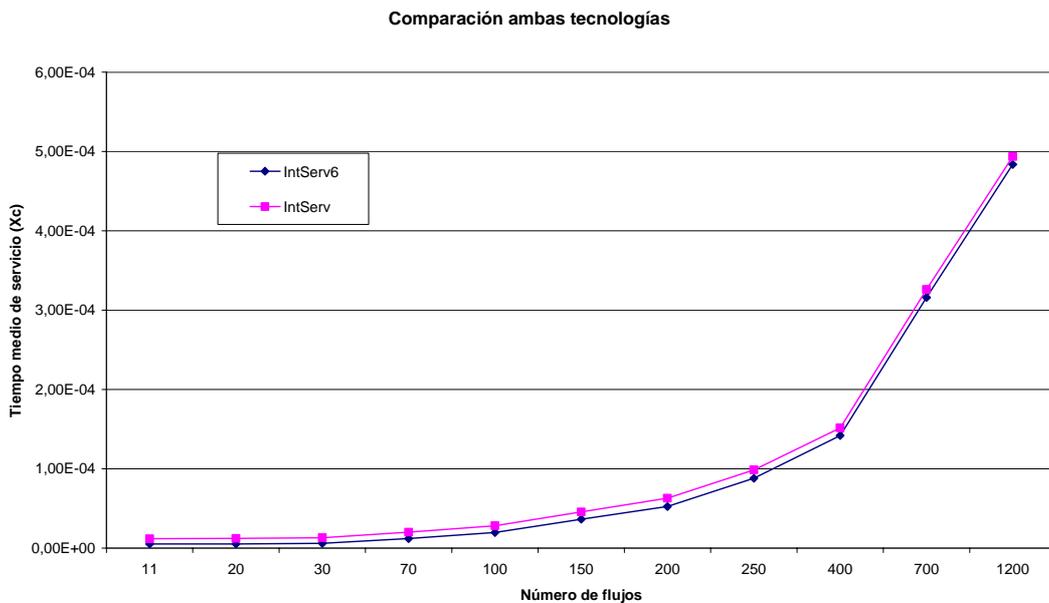


Figura 4.17 Comparación de las simulaciones del *Tiempo Medio de Clasificación* en *IntServ* e *IntServ6*.

4.3.3.2 Simulaciones del retardo medio

Además de simular el *Tiempo de Servicio*, se realizaron simulaciones donde se obtienen las gráficas para el *Tiempo de Retardo* de los paquetes con diferente *Número de Flujos* en el *Encaminador*. Estos resultados se observan en la Figura 4.18, donde es claro que la tecnología *IntServ6* posee unos *Tiempos de Retardo* mucho menores que los *Tiempos de Retardo* de *IntServ*. También se observa que el *Tiempo de Retardo* de los paquetes para *IntServ* crece mucho más rápido que *IntServ6* a partir de los 200 flujos aproximadamente, lo que indica un deterioro de la *Calidad de Servicio* a partir de esta cantidad de flujos. Por otro lado, para el caso de *IntServ6*, el retardo de paquetes se mantiene en niveles mucho más bajos que los de *IntServ* incluso cerca de los 1000 flujos. Sin embargo, al alcanzar los 1200 flujos, el retardo de clasificación en *IntServ6* crece drásticamente. Estas observaciones permiten concluir que *IntServ* mantiene un retardo aceptable hasta los 200 flujos, mientras que *IntServ6* mantiene un retardo moderado hasta un valor por debajo de los 1200 flujos. Esto se debe a que para un *Número de Flujos* superior a estos valores, la *Tasa Media de Servicio* es menor que la *Tasa Media de Llegadas* y la *Longitud de la Cola* se incrementa drásticamente.

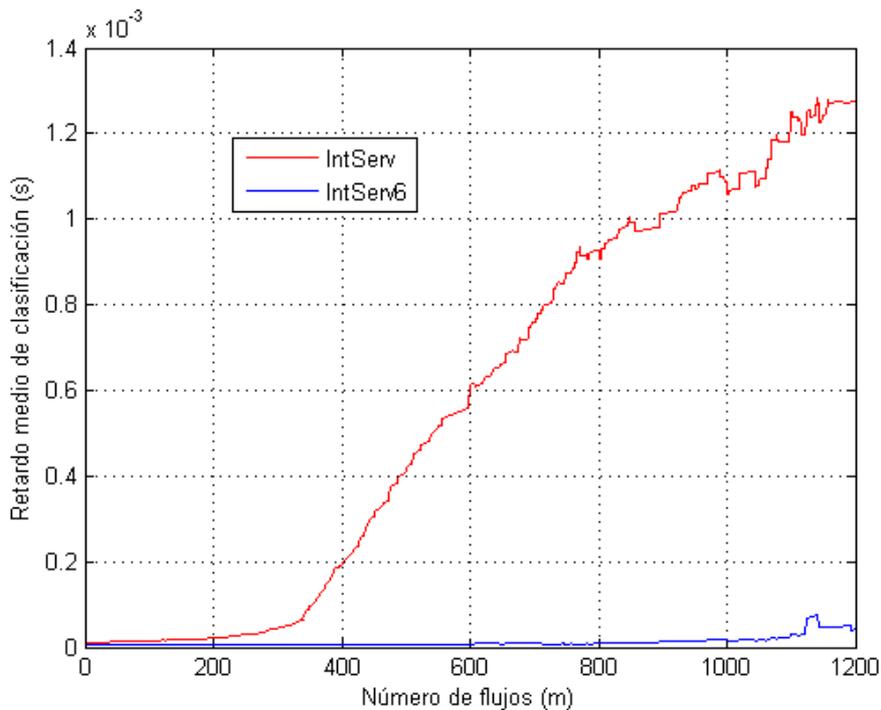


Figura 4.18 Simulación del *Retardo* de los paquetes en el *Clasificador*.

Estos resultados coinciden con los del modelo teórico del retardo (ver Figura 4.19), donde se observa que la asíntota del crecimiento del retardo, tanto para *IntServ* como para *IntServ6*, está ubicada en valores muy similares a los valores obtenidos mediante simulación.

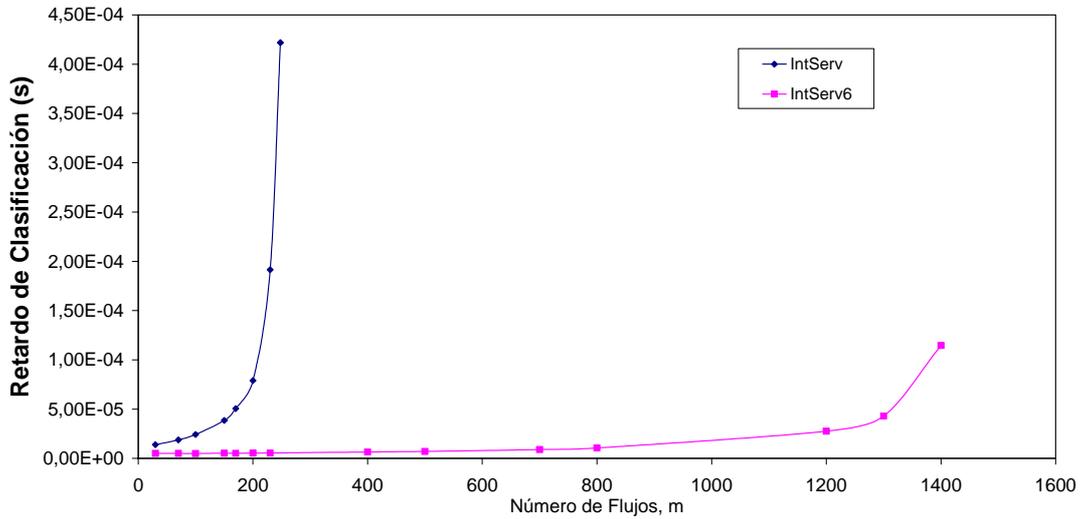


Figura 4.19 Resultados del *Retardo* de los paquetes para el modelo teórico.

Un acercamiento de los *Tiempos de Retardo* para menos de 100 *Flujos* en las simulaciones (ver Figura 4.20, que se encuentra en escala logarítmica en el eje vertical), permite observar que el *Tiempo Medio de Retardo* en este rango tiene una diferencia de dos ordenes de magnitud para las dos tecnologías. Es decir, el retardo de clasificación de *IntServ6* es aproximadamente 100 veces menor que el retardo de clasificación de *IntServ*. Así, *IntServ6* tiene un mejor desempeño con respecto a *IntServ*, lo que se debe a que el cálculo del *Número Hash* es realizado en el ordenador del usuario final y no en el *Encaminador*. Además, en *IntServ* el *Encaminador* debe realizar este cálculo por cada paquete que llega.

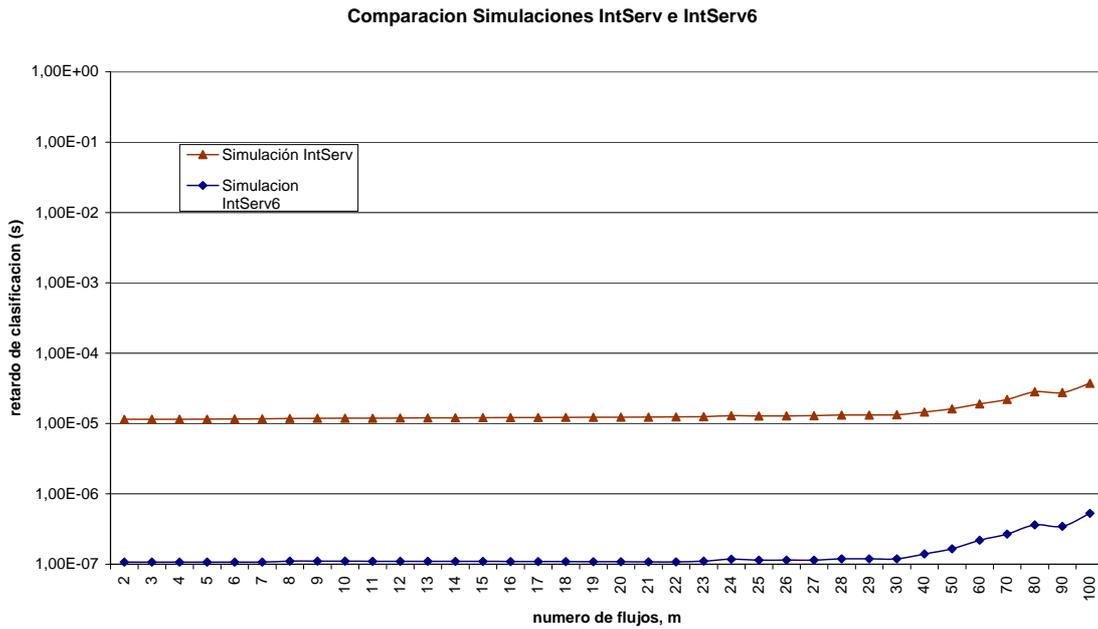


Figura 4.20 Comparación del *Retardo Medio* de paquete en ambas tecnologías para menos de 100 flujos

4.4 Evaluación de las reservas para IntServ6

En secciones anteriores se realizó el estudio del principal parámetro que se garantiza en una red de *Servicios Integrados*, el retardo de los paquetes. Sin embargo, otro de los principales parámetros que debe analizarse cuando hablamos de reservas en *Encaminadores de Servicios Integrados* es el *Ancho de Banda Real* asignado a cada reserva. Este puede medirse a partir del *Tráfico Servido* para cada sesión (o flujo) que, a su vez, es un valor que al ser evaluado en un período de tiempo debería ofrecer un valor cercano al ancho de banda al que se ha comprometido la red cuando se establece la reserva.

En este sentido y debido a que *IntServ6* propone un cambio en el proceso de clasificación pero no cambia el proceso de planificación de los *Encaminadores IntServ* (lo que quiere decir que se sigue utilizando un planificador de tipo *GPS*), debe estudiarse el efecto sobre el *Tráfico Servido* debido al cambio en el clasificador. Para hacer este análisis se parte del siguiente escenario:

- El *Clasificador* y el *Planificador* se encuentran organizados en *Tándem*, de forma que para cada paquete primero se realiza la *Clasificación* y luego la *Planificación*.
- El *Planificador* no ha alcanzado su máxima capacidad, por lo que podría gestionar un mayor tráfico de entrada si es necesario.

Bajo estas condiciones, si se disminuye el *Tiempo de Servicio* del *Clasificador*, se incrementará la *Tasa de Llegada de Paquetes* al *Planificador*. En este caso, si así lo requieren las reservas de los flujos, el *Planificador* podría incrementar el tráfico de salida de los diferentes flujos.

Como un ejemplo simple, en una cola de tipo *FIFO* (*First-in First-Out*) con servicio determinista (la cual brinda igual tratamiento a cada paquete sin importar a qué flujo pertenezca), se podría incrementar el *Tráfico Servido* simplemente disminuyendo el *Tiempo de Servicio* por paquete. De esta forma todos los flujos recibirían el mismo incremento en el *Tráfico Servido*. Sin embargo, un planificador *GPS*, aunque simple en su principio básico, presenta aspectos complejos de modelar para obtener expresiones que permitan describir parámetros estocásticos tales como el *Retardo Medio* de los paquetes, la *Longitud Media de Cola*, el *Tráfico Servido*, etc. Es por esta razón que es necesario establecer el comportamiento del *Tráfico Servido* para cada flujo en un planificador *GPS* para así determinar cómo cambiaría este parámetro al variar el *Tráfico Entrante* y determinar así la variación en el *Tráfico de Salida* del *Encaminador* para cada uno de los flujos. Por tanto, se ha realizado un estudio del estado del arte de los modelos estocásticos de los planificadores *GPS*. Inicialmente, Parekh y Gallager en [36, 88] han hecho estudios acerca de cuál es el *Peor Retardo*, en un planificador *WFQ* y en una red de planificadores en cascada respectivamente. Desafortunadamente, este parámetro nos da una idea del peor caso, pero no permite dar una idea de los valores medios de *Retardo* y *Ancho de Banda*, los cuales son muy importantes a la hora de especificar los servicios de una red de *Servicios Integrados*. Es así, como Yaron y Sidi en [89] plantean un método para caracterizar de alguna manera un sistema tan complejo. La teoría desarrollada por estos autores utiliza ráfagas de paquetes limitadas exponencialmente y se conoce como *EBB* (*EBB- Exponentially Bounded Burstiness*). Esta teoría [90] plantea que cuando las entradas a un elemento de red aislado están todas

limitadas, sus resultados serán salidas limitadas, y se asegura que los *Retardos* y las *Longitudes de las Colas* en este elemento tengan distribuciones con decaimiento exponencial. Adicionalmente, en algunos tipos de redes, las entradas limitadas resultan en salidas limitadas y puede demostrarse que los procesos naturales de tráfico satisfacen tales límites. En consecuencia, este método permite el análisis de diferentes casos anteriormente intratables. En esta teoría, se proveen suficientes condiciones para la estabilidad de tales redes, y se derivan *Límites Superiores* para los parámetros de interés del desempeño de las redes.

Por otro lado, Zhang y Towsley en [91] utilizan el modelo de procesos *EBB* introducido por Yaron para obtener resultados que extienden los resultados determinísticos de Parekh y Gallager mencionados anteriormente. Es así, como Zhang y Towsley presentan *Límites Estadísticos* en las distribuciones de la *Longitud de la Cola* y el *Retardo de Paquetes* en cola para el caso de planificadores *GPS*. En este estudio, también se obtienen expresiones para el *Límite Superior del Tráfico Servido* para cualquier sesión (flujo) servido por el *Planificador*.

Ya que el *Tráfico Servido* permite obtener una medida del *Ancho de Banda* servido para una determinada sesión, en esta sección se aplicarán los resultados obtenidos por Zhang en [91] para analizar cómo se afecta el *Límite Superior del Tráfico Servido* bajo diferentes situaciones tales como *Reservas Simples* y *Reservas en Túneles* bajo las dos tecnologías de *Servicios Integrados: IntServ* e *IntServ6*.

4.4.1 Efecto del tiempo de clasificación en el tráfico servido

A continuación, se describirá el modelo propuesto por Zhang y Towsley en [91] para describir el *límite superior del tráfico servido* en un planificador *GPS*. Una vez planteado este modelo, se describirá como se afecta este límite superior debido al *tiempo de clasificación* de los paquetes.

4.4.1.1 El modelo de Zhang

El modelo de Zhang puede entenderse si se observa la Figura 4.21. En esta figura se muestra el comportamiento del *tráfico de llegada* y el *tráfico servido* para el *i*-ésimo flujo en un planificador *GPS*.

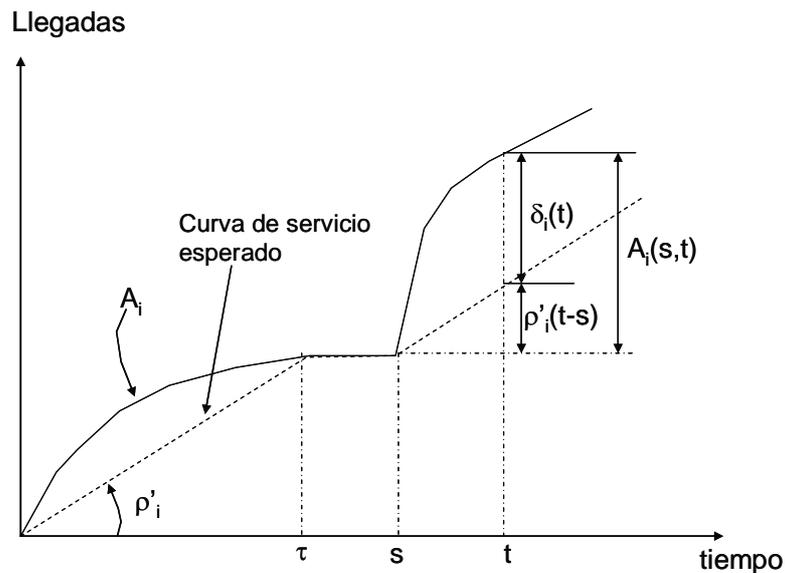


Figura 4.21 Comportamiento del *tráfico de llegada* y el *tráfico servido* para el flujo *i*-ésimo en un planificador *GPS*.

Las variables que aparecen en este modelo son las siguientes:

A_i : Cantidad de tráfico que llega del flujo *i*.

ρ'_i : *Tasa de servicio esperada*. A esta tasa se compromete el planificador para la sesión *i*.

S_i : Cantidad de tráfico servido del flujo *i*.

$\rho'_i(t-s)$: Cantidad de tráfico servida entre los instantes *s* y *t*.

$A_i(s,t)$: Cantidad de tráfico de llegada entre los instantes *s* y *t*.

$\delta_i(t)$: Longitud esperada de la cola para el flujo *i* (ó Tráfico esperado en cola para el flujo *i*).

El modelo matemático está basado en el concepto de *ordenamiento factible* y contempla entre otras cosas, la expresión para el *límite superior del tráfico servido* para el flujo i entre los instantes τ y t . El concepto de *ordenamiento factible* fue introducido por Parekh y Gallager [36, 88]. Estos demostraron que, si ρ_i es la *tasa superior de largo plazo* del proceso de llegadas del flujo i y, dado que $\sum_{i=1}^N \rho_i < 1$, existe un ordenamiento entre las sesiones tal que, después de re-etiquetar las sesiones se cumple que,

Ecuación 4.56:

$$\rho_i < \frac{\phi_i}{\sum_{j=1}^N \phi_j} (1 - \sum_{j=1}^{i-1} \rho_j), \quad 1 \leq i \leq N$$

Donde ϕ_i es el *peso* que se le asigna a el flujo i -ésimo y N es el número total de flujos siendo atendidos por el servidor. Tal ordenamiento es llamado *ordenamiento factible* (con respecto a $\{\rho_i\}_{1 \leq i \leq N}$ y $\{\phi_i\}_{1 \leq i \leq N}$).

En consecuencia, para obtener la expresión del límite superior del tráfico servido para un flujo i , en [91] se parte de que,

Ecuación 4.57:

$$S_i(\tau, t) \leq A_i(\tau, t) + Q_i(\tau)$$

Donde $Q_i(\tau)$ es la *cantidad real de tráfico en cola* para el flujo i en el instante τ . Y si además se tiene que,

Ecuación 4.58:

$$\sum_{j=1}^i Q_j(t) \leq \sum_{j=1}^i \delta_j(t)$$

Es decir que, en un *ordenamiento factible*, el acumulado del *tráfico real en cola* de las primeras i sesiones nunca excede el *tráfico esperado en cola* del agregado de las mismas i sesiones. Bajo estas condiciones en [91] se demuestra que,

Ecuación 4.59:

$$Q_i(t) \leq \delta_i(t) + \frac{\phi_i}{\sum_{j=1}^N \phi_j} \sum_{j=1}^{i-1} \delta_j(t)$$

Donde el término, $\frac{\phi_i}{\sum_{j=1}^N \phi_j}$ es la *tasa garantizada de vaciado del tráfico en cola*, por lo

que el término $\frac{\phi_i}{\sum_{j=1}^N \phi_j} \sum_{j=1}^{i-1} \delta_j(t)$ será la proporción del tráfico a servir de las restantes

$(i-1)$ sesiones a la *tasa garantizada de vaciado de tráfico de la cola*. Por tanto, reemplazando la Ecuación 4.59 en la Ecuación 4.57 se puede expresar el *Límite Superior del Tráfico Servido* para el flujo i como,

Ecuación 4.60:

$$S_i(\tau, t) \leq A_i(\tau, t) + \delta_i(\tau) + \frac{\phi_i}{\sum_{j=i}^N \phi_j} \sum_{j=1}^{i-1} \delta_j(\tau)$$

4.4.1.2 Efecto de la clasificación en el tráfico servido

En las tecnologías que se desea comparar para el soporte de *Calidad de Servicio*, es decir, *IntServ* e *IntServ6*, la principal diferencia en la arquitectura de los *Encaminadores* se presenta en el proceso de *clasificación*, mientras que el proceso de *planificación* es el mismo. Por tanto, debido a que no hay cambios en el proceso de *planificación*, no hay efectos directos debidos al uso de diferentes planificadores sobre el *ancho de banda servido*.

Sin embargo, se debe tener en cuenta que al cambiar el proceso de *clasificación*, el *tiempo medio de servicio* de este proceso cambia. A su vez, el cambio en el *tiempo medio de clasificación* produce un cambio sobre la *cantidad de tráfico de llegada al planificador*. Como ejemplo, si se llegase a disminuir el *tiempo de clasificación*, la *tasa media de llegada de paquetes al planificador* se incrementaría. Este incremento sería el mismo para los paquetes de todos los flujos debido a que el clasificador da el mismo tratamiento a los paquetes independientemente del flujo al que pertenezcan.

En este sentido y debido a que *IntServ6* propone un cambio en el proceso de clasificación pero no cambia el proceso de planificación de los *Encaminadores IntServ* (lo que quiere decir que se sigue utilizando un planificador de tipo *GPS*), debe

estudiarse el efecto sobre el *tráfico servido* debido al cambio en el clasificador. Para hacer este análisis se parte del siguiente escenario:

- El clasificador y el planificador se encuentran organizados en *Tándem*, de forma que para cada paquete primero se realiza la clasificación y luego la planificación.
- El planificador no ha alcanzado su máxima capacidad, por lo que podría gestionar un mayor tráfico de entrada si es necesario.

Bajo estas condiciones, si se disminuye el tiempo de servicio del clasificador, se incrementará la tasa de llegada de paquetes al planificador. En este caso, si así lo requieren las reservas de los flujos, el planificador podría incrementar el tráfico de salida de los diferentes paquetes.

Suponiendo que la proporción en que se incrementa la *tasa de llegada de paquetes* es p , podemos decir que en media el *tráfico de entrada* del flujo i entre los instantes τ y t se incrementará en p también; esto es,

Ecuación 4.61:

$$A'_i(\tau, t) = p A_i(\tau, t)$$

De la misma manera, podemos asumir que, si no se cambiase el *tiempo de servicio* para ningún flujo, la cantidad de tráfico de las colas de todos los flujos se incrementa en la misma proporción p . Por tanto, obtenemos que,

Ecuación 4.62:

$$\delta'_i(\tau) = p \delta_i(\tau)$$

De esta forma, la expresión para el límite superior de la *cantidad de tráfico servido* para las nuevas condiciones de clasificación será,

Ecuación 4.63:

$$S'_i(\tau, t) \leq p \cdot A_i(\tau, t) + p \cdot \delta_i(\tau) + \frac{\phi_i}{\sum_{j=i}^N \phi_j} p \cdot \sum_{j=1}^{i-1} \delta_j(\tau)$$

Si se compara la Ecuación 4.63, que representa el *Límite Superior del Tráfico Servido* al incrementar el *tráfico de entrada* hasta $A'_i(\tau, t)$, con la Ecuación 4.60, que representa el *límite superior del tráfico servido* para el tráfico de entrada inicial, se obtiene que el *límite superior del tráfico servido* se ve afectado en la misma proporción p en que se modifica el *tiempo medio de clasificación*. De esta manera queda demostrado que al aumentar el *tráfico de entrada* de todos los flujos en una misma proporción p , se obtiene un aumento en el *límite superior del tráfico servido* en un planificador GPS. Esto quiere decir que, siempre y cuando el planificador no hubiese alcanzado su máxima capacidad, este podría aumentar su tasa de servicio para el flujo i hasta el nuevo límite dado por la Ecuación 4.63.

Por otra parte, el valor de p es en general un cociente cuyo cálculo depende de la situación que estemos evaluando. Existen dos situaciones que deseamos evaluar para comparar las tecnologías *IntServ* e *IntServ6*: (a) la variación del *tráfico servido* para un flujo i en caso de *reservas simples*, y (b) la variación del *tráfico servido* para un flujo i en caso de *reservas en túneles*. A continuación se explicará el cálculo de p para cada una de estas situaciones.

4.4.1.3 Variación del tráfico servido en caso de reservas simples

A continuación se calculará la proporción de reducción del tiempo entre llegadas al planificador para el caso de *reservas simples*. Esta proporción de reducción toma como referencia el *tiempo medio entre llegadas* de *IntServ*, el cual a su vez es el *tiempo medio de servicio del clasificador* (\bar{x}_{co}). Este se calcula mediante la Ecuación 4.29, obtenida en la sección 4.3.2.5. Por tanto, la expresión para calcular la proporción de la reducción en tiempo medio de llegadas se calcula como,

Ecuación 4.64:

$$r = \frac{\bar{x}_c}{\bar{x}_{co}}$$

Donde \bar{x}_c : Tiempo medio de clasificación para *IntServ6*, dado por la Ecuación 4.18

\bar{x}_{co} : Tiempo medio de clasificación para *IntServ*, dado por la Ecuación 4.29

De igual manera, podemos expresar la proporción de aumento en la tasa de llegadas de *IntServ6* con respecto a *IntServ* como el inverso de la Ecuación 4.64,

Ecuación 4.65:

$$a = \frac{\bar{x}_{co}}{\bar{x}_c}$$

Así, el valor de p (ganancia o proporción de incremento de llegadas de paquetes), lo podemos expresar como,

Ecuación 4.66:

$$p = a = \frac{\bar{x}_{co}}{\bar{x}_c}$$

4.4.1.4 Variación del tráfico servido en caso de reservas en túneles

Para el caso de reservas en túneles, debe tenerse en cuenta que el *tiempo de clasificación* cambia para la tecnología *IntServ* estándar, mientras que para *IntServ6* sigue siendo el mismo. Como se explicó en la sección 3.3, las reservas en túneles para *IntServ6* se siguen manejando a nivel de *capa de red* mediante *etiquetas*, por lo que el proceso de clasificación en los *Encaminadores* dentro del túnel permanece inalterado. Por otro lado, para el caso de *IntServ*, el proceso de clasificación debe analizar información de la *capa de transporte*. Este procesamiento adicional aumenta el tiempo de clasificación. El proceso completo para un *Encaminador interno de un túnel* en *IntServ* se observa en la Figura 4.22.

A partir de la Figura 4.22, se puede observar que la diferencia entre este proceso y el proceso de clasificación durante una *reserva simple* (no perteneciente a un túnel) es que para el caso de reservas en túneles se agrega un proceso adicional de *análisis del puerto fuente* para diferenciar la *reserva dentro del túnel*. Por tanto, se debe calcular el tiempo adicional debido a este proceso.

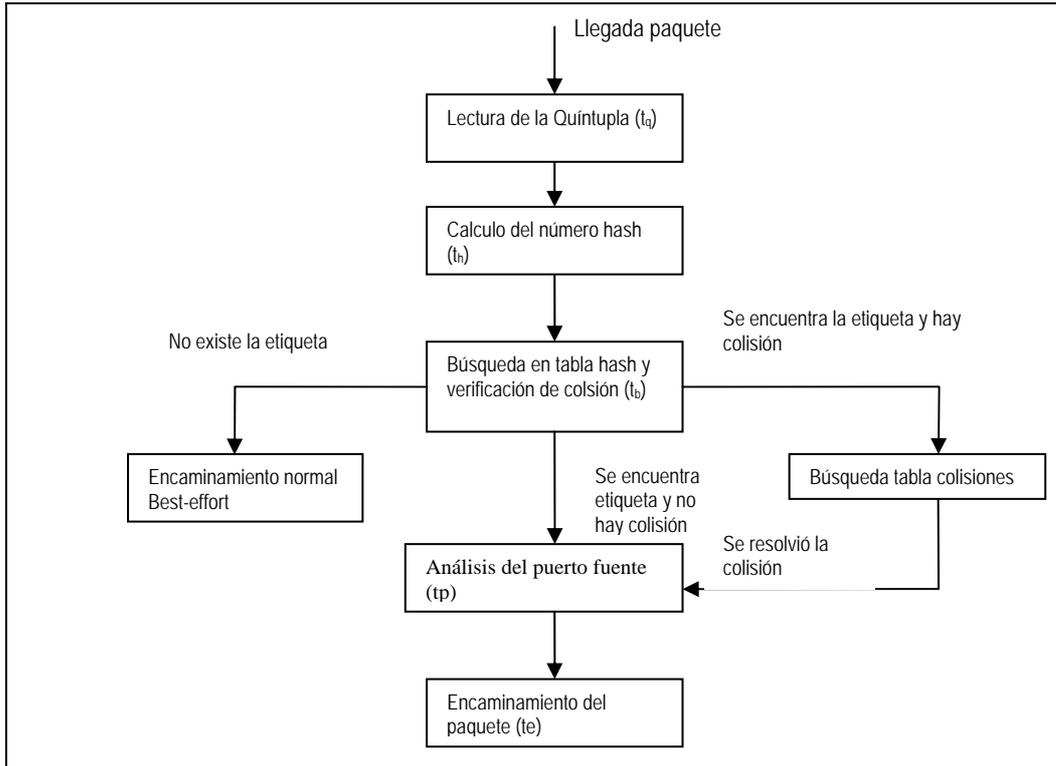


Figura 4.22 Diagrama de estados de un *Encaminador IntServ* para procesar reservas dentro de un túnel

El proceso de *análisis del puerto fuente* consiste esencialmente de dos etapas: la primera es la lectura del *puerto fuente* de la cabecera *UDP*, mientras que la segunda es la búsqueda de la reserva en una *Tabla de Reservas del túnel*. Para el caso de la lectura del *puerto fuente* de la cabecera *UDP*, se asume que esta lectura ha sido realizada previamente durante la lectura de la *Quíntupla*, por lo que no aporta un tiempo adicional de procesamiento. Sin embargo, la etapa de búsqueda de la reserva en la *Tabla de Reservas en túneles* sí hace un aporte importante al *tiempo de servicio* de los paquetes en el *Encaminador*. Para calcular este tiempo, partimos de que se hace una búsqueda en la tabla de tipo *matching*. Esto implica que la probabilidad de éxito tiene una *distribución geométrica*, similar al caso descrito para la *Tabla de resolución de colisiones* (ver secciones 4.3.2.4 y 4.3.2.5, modelo del retardo en la clasificación de paquetes). Por tanto, el *tiempo medio de búsqueda* viene determinado por el *número medio de lecturas de la tabla* (lecturas en *SRAM*). El *número medio de intentos* se obtiene para una *distribución geométrica* como $(1/p_{\text{éxito}})$, por lo que el tiempo medio de búsqueda será,

Ecuación 4.67:

$$\bar{T}_{bt} = \frac{1}{p_{\text{exito}}} T_{SRAM}$$

Donde,

Ecuación 4.68:

$$p_{\text{exito}} = \frac{1}{m_{\text{tunel}}}$$

Y por tanto,

Ecuación 4.69:

$$\bar{T}_{bt} = m_{\text{tunel}} T_{SRAM}$$

Donde el valor de m_{tunel} es el *número de reservas* que hay dentro del túnel. Finalmente, el tiempo medio total de proceso de la clasificación de un paquete en un *Encaminador interno* de un túnel viene dado por el valor del *tiempo de clasificación* en una reserva simple \bar{x}_{co} más el tiempo adicional \bar{T}_{bt} . Entonces, la expresión del *tiempo medio total de clasificación* para una *reserva del túnel* será,

Ecuación 4.70:

$$\bar{T}_t = \bar{x}_{co} + m_{\text{tunel}} T_{SRAM}$$

Finalmente, se calculará la proporción de reducción del tiempo entre llegadas al planificador con respecto a *IntServ* como,

Ecuación 4.71:

$$r = \frac{\bar{x}_c}{\bar{T}_t}$$

Por tanto, la proporción de aumento en la tasa de llegadas al planificador con respecto a *IntServ* será entonces,

Ecuación 4.72:

$$a = \frac{1}{r} = \frac{\bar{T}_t}{\bar{x}_c}$$

Por lo que, p (ganancia o proporción de incremento en la llegada de paquetes) será,

Ecuación 4.73:

$$p = a = \frac{\bar{x}_{co} + m_{tunnel} T_{SRAM}}{\bar{x}_c}$$

4.4.2 Resultados Obtenidos

En primer lugar, se analizan los resultados de la *ganancia* (p) en el límite superior del tráfico servido para el caso de un *Encaminador con reservas simples*. En esta situación, se varía el *número de flujos* y se obtiene la Figura 4.23. En ella se observa que la *ganancia* para el tráfico servido para *IntServ6* con respecto a *IntServ* aumenta en la medida que el *número de flujos* aumenta. Este fenómeno se debe a que el tiempo de clasificación para *IntServ* aumenta drásticamente con el *número de flujos*, mientras que el tiempo de clasificación para *IntServ6* aumenta muy lentamente. Como consecuencia, al aumentar el número de flujos que tienen reserva en el *Encaminador*, el límite superior de tráfico servido por *IntServ* empeora de una manera más drástica que el tráfico servido en *IntServ6*.

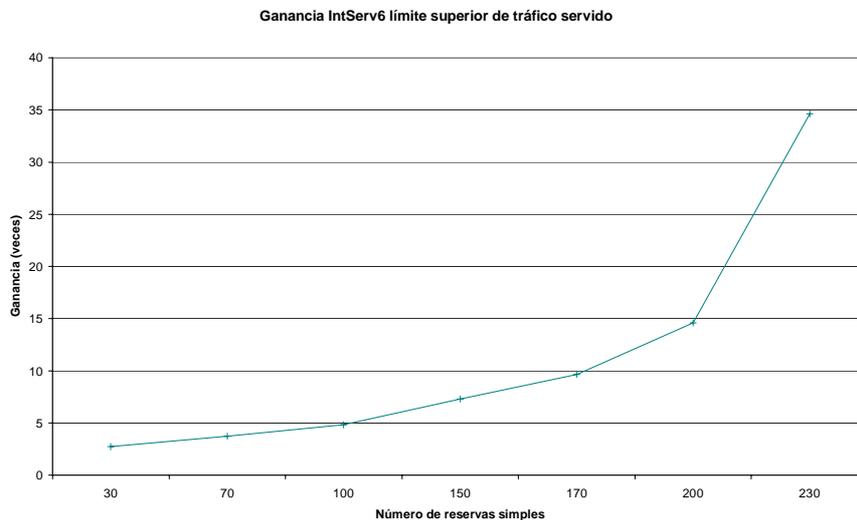


Figura 4.23 Ganancia en el *límite superior del tráfico servido* cuando se varía el *Número de Flujos* en reservas simples.

Ahora, se analizará el caso de la *ganancia* en el límite superior del tráfico servido para el caso de un *Encaminador con reservas en túneles*. Para esta situación se estudia el

efecto de cambiar el *número de reservas dentro del túnel*, ya que según la Ecuación 4.73, la ganancia p depende de este parámetro. Por esta razón, las condiciones de simulación implican un número de *reservas simples* constante ($m=200$), por lo que los tiempos de clasificación \bar{x}_{co} y \bar{x}_c permanecen constantes ($\bar{x}_{co}=7,89644 \times 10^{-05}$ s y $\bar{x}_c=5,411003843 \times 10^{-06}$ s). Bajo estas condiciones, se varía el número de reservas en el túnel de 1 a 200. Los resultados se observan en la Figura 4.24.

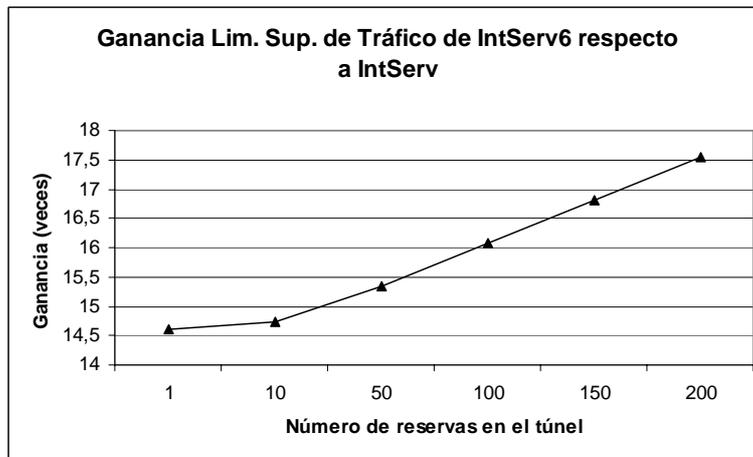


Figura 4.24 Ganancia de *IntServ6* con respecto a *IntServ* en el *límite superior de tráfico servido*.

Podemos observar en la Figura 4.24 que hay un aumento en la *ganancia* que es proporcional al *número de flujos* que contiene el túnel. Sin embargo, el hecho de que la ganancia de *IntServ6* con respecto a *IntServ* en el *límite superior del tráfico servido* aumente, no quiere decir que el *límite superior* de *IntServ6* aumente, lo que sucede realmente es que el *límite superior del tráfico servido* para *IntServ6* permanece constante ya que su tiempo de clasificación permanece constante, tal como se observa en la Tabla 4.6, mientras que el *límite superior de tráfico servido* para *IntServ* se va desmejorando en la medida que hay más reservas dentro del túnel debido a que su tiempo de clasificación va aumentando.

Tabla 4.6 Valores de tiempo medio de clasificación de las reservas en túneles

Número de flujos con reservas simples	Número de reservas dentro del túnel	Tiempo de clasificación <i>IntServ</i> (segundos)	Tiempo de clasificación <i>IntServ6</i> (segundos)
200	1	$7,9 \times 10^{-5}$	$5,41 \times 10^{-6}$
200	10	$7,97 \times 10^{-5}$	$5,41 \times 10^{-6}$
200	50	$8,29 \times 10^{-5}$	$5,41 \times 10^{-6}$
200	100	$8,69 \times 10^{-5}$	$5,41 \times 10^{-6}$
200	150	$9,09 \times 10^{-5}$	$5,41 \times 10^{-6}$
200	200	$9,49 \times 10^{-5}$	$5,41 \times 10^{-6}$

La Figura 4.25 confirma este aspecto. En ella observamos que la *ganancia* en el límite superior del tráfico servido de *IntServ* con respecto a *IntServ6* es menor que 1 y va bajando en la medida que el *número de flujos* dentro del túnel aumenta. En realidad, lo que se tiene en *IntServ* es una pérdida en el tráfico servido proporcional al *número de flujos* dentro del túnel.

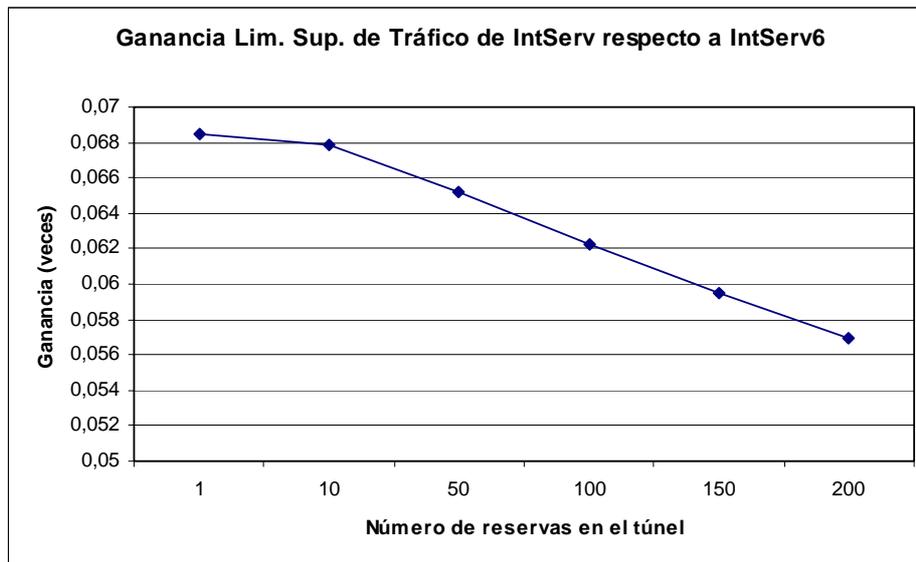


Figura 4.25 Ganancia de *IntServ* con respecto a *IntServ6* en el límite superior de tráfico servido.

4.5 Dependencia entre el tiempo de retardo y la movilidad de los usuarios

Hasta ahora, se han realizado diferentes estudios sobre el efecto de la movilidad en esquemas de *Control de Acceso (CAC)*[99-101]. Dichos trabajos han obtenido expresiones a nivel del desempeño de llamadas con parámetros tales como la *Probabilidad de Bloqueo de Nuevas Llamadas* y la *Probabilidad de Interrupción de Llamadas de Traspaso*. Sin embargo, las redes celulares de próxima generación estarán basadas en conmutación de paquetes. Por tanto, es muy importante calcular parámetros como el retardo de los paquetes. Recientemente, en [102] se presentó una propuesta para modelar el *Retardo de los Paquetes* y la *Probabilidad de Llenado de los Buffers*. Dicha propuesta planteó un modelo muy simple de una estación base inalámbrica y para ello utilizó un modelo general de un multiplexor. Sin embargo, es necesario obtener modelos más complejos que permitan a los diseñadores de redes describir estos parámetros de una mejor manera.

El modelo obtenido en el presente trabajo fue presentado en [10, 11] y describe el efecto de la movilidad sobre el retardo medio de los paquetes en un *Encaminador de Acceso Inalámbrico* basado en *Servicios Integrados* para el soporte de *QoS*. Este modelo es utilizado para evaluar el efecto de la movilidad de los usuarios sobre dos tecnologías de *QoS*: *IntServ* e *IntServ6*.

4.5.1 Antecedentes

Como se ha mencionado, hay diferentes trabajos que han estudiado los efectos de la movilidad sobre los esquemas de *Control de Admisión (CAC)* en redes celulares [99-101]. Algunos de estos trabajos han estudiado ciertos esquemas *CAC* particulares y asumen que los *Tiempos de Retención del Canal (Channel Holding Time)* para *Llamadas Nuevas* y *Llamadas de Traspaso* son independientes e idénticamente distribuidos de acuerdo a una *Distribución Exponencial*. Pero diferentes estudios de las redes de datos han demostrado que el tiempo de retención del canal no necesariamente sigue una *Distribución Exponencial*. En una serie de estudios [103-105], Fang y otros

demonstraron que el *tiempo de retención* del canal para *Nuevas Llamadas* y para *Llamadas de Traspaso* pueden tener diferentes distribuciones y diferentes valores medios. Sin embargo, en [105] Fang hace una nueva aproximación para evitar tener que solucionar un gran conjunto de ecuaciones de flujo. En esta aproximación, las *Llamadas Nuevas* y las *Llamadas de Traspaso* se consideran independientes y exponencialmente distribuidas pero con diferentes valores medios. El análisis y los resultados de simulación muestran que la aproximación de Fang [105] es más realista y se comporta mucho mejor que la aproximación tradicional. Por tanto, el modelo propuesto en el presente trabajo utiliza dicha aproximación.

4.5.2 Modelo del Encaminador de Acceso inalámbrico

El modelo utilizado es similar al propuesto para el caso de *Encaminadores* fijos y se observa en la Figura 4.26. Un *Encaminador de Servicios Integrados* está compuesto principalmente por dos partes; la primera parte es el *gestor de las reservas*, el cual está compuesto por el *Esquema de Control de Admisión (CAC)* y el *Agente de Establecimiento de las Reservas*. La segunda parte del *Encaminador* realiza el proceso de retransmisión de paquetes y está compuesta por el proceso de *Clasificación* y el proceso de *Planificación* de paquetes. Como se observa en la Figura 4.26, estas dos partes principales están relacionadas entre sí a través de la *Tabla de Reservas de Flujos*. Por tanto, cuando el módulo *CAC* acepta una *Nueva Llamada* (o un nuevo flujo, ya que en este contexto llamada y flujo tienen la misma connotación), el *número de flujos* en la *Tabla de Reservas* se incrementa en uno. Como resultado, cuando llega un nuevo paquete, este debe ser clasificado, es decir, el paquete debe ser buscado en la *Tabla de Reserva de flujos*. A continuación, si el paquete pertenece a algún flujo, deberá ser enviado al *Planificador* del enlace de salida correspondiente. En este modelo (al igual que para el *Encaminador fijo*), para simplificar, suponemos que el *tiempo de retardo* del módulo de *Control de Políticas* es cero.

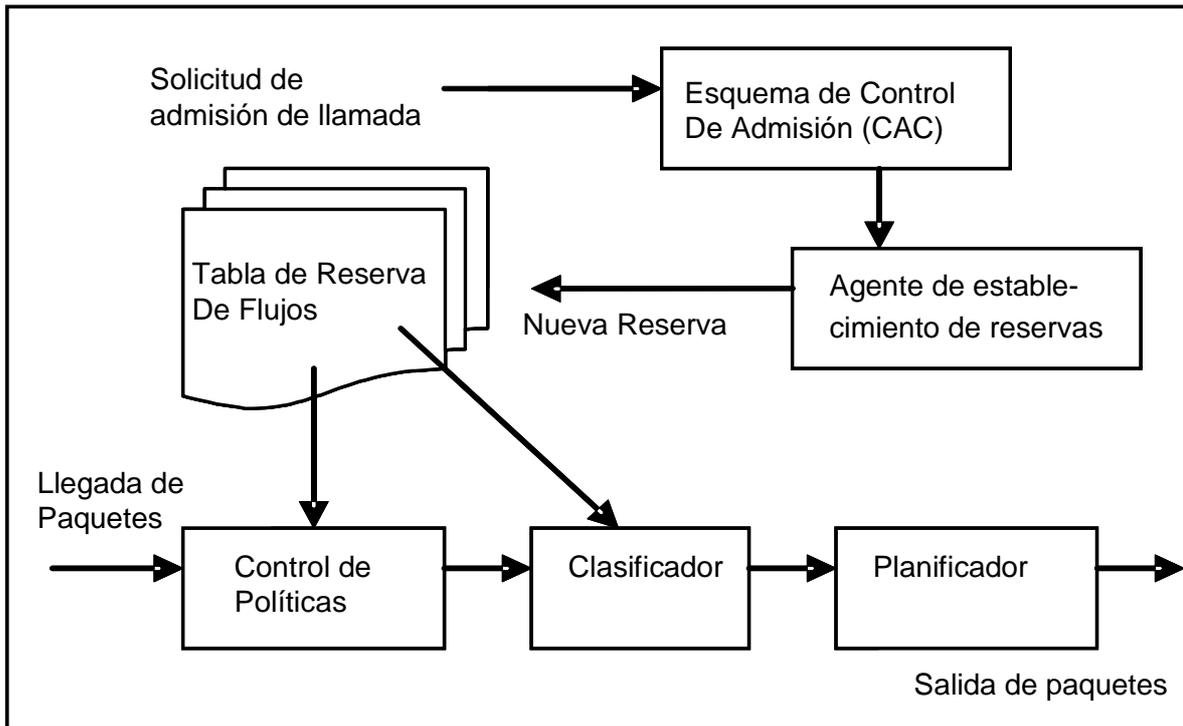


Figura 4.26 Modelo de un *Encaminador de Acceso Inalámbrico*.

Con base en el modelo descrito, se ha desarrollado un modelo matemático que describe diferentes características. La primera característica es la relación entre N_t (el *Número de Flujos* que la *Tabla de Reserva de flujos* almacena), ρ_n (la *Intensidad de Tráfico de Nuevas Llamadas*) y ρ_h (la *Intensidad de Tráfico de Llamadas de Traspaso*). La segunda característica modelada es el *Retardo Medio de las Ráfagas de Paquetes* en el *Clasificador*. Este parámetro depende del *Número de Flujos* (o llamadas) que contiene la *Tabla de Reservas de Flujos*. Un tercer aspecto modelado es el *Retardo Medio* de los paquetes para el proceso de *Planificación*.

Para obtener estos parámetros, se parte de los mismos supuestos que se plantearon para el modelo de colas del *Encaminador Fijo*. Esto es, para simplificar el análisis del proceso de clasificación, se supone que los paquetes llegan a ráfagas y, al mismo tiempo, tales ráfagas llegan con una *Distribución de Poisson*. También se supone que el *Encaminador* tiene una *Memoria Temporal* que almacena cada ráfaga, de forma que sólo debe realizar el proceso de clasificación una vez por cada ráfaga de paquetes. Estos supuestos son posibles con base en los estudios realizados por Roberts en [87], donde se plantea que tal comportamiento puede ser obtenido del tráfico de *Internet*

generado a partir de conexiones *TCP* y aplicaciones tales como el *Web* y la *transferencia de ficheros (FTP- File Transfer Protocol)*.

4.5.3 Modelo de Movilidad y esquemas de Control de Admisión

Como se ha mencionado antes, el modelo planteado utiliza el modelo de movilidad de Fang descrito en [105]. Este modelo asume que los procesos de llegadas de *Nuevas Llamadas* y *Llamadas de Traspaso* son de tipo *Poisson*. Los tiempos de retención del canal para las *Nuevas Llamadas*, y para las *Llamadas de Traspaso*, se asumen independientes y exponencialmente distribuidos pero con diferentes valores medios. Sean λ_n , λ_h , $1/\mu_n$ y $1/\mu_h$ las *tasas de llegada de Nuevas Llamadas* y de *Llamadas de Traspaso* y los valores medios de *tiempos de retención del canal para Llamadas Nuevas* y de *Traspaso* respectivamente. Sean $\rho_n = \lambda_n/\mu_n$ y $\rho_h = \lambda_h/\mu_h$. Sea C el número total de canales en una celda. El modelo de movilidad de Fang obtiene para diferentes esquemas CAC los parámetros de desempeño a nivel de llamadas tales como: *Probabilidad de Bloqueo de Nuevas Llamadas* (P_{nb}) y *Probabilidad de Interrupción de Llamadas de Traspaso* (P_{hb}). Los esquemas CAC modelados en [105] y que son usados en nuestro modelo se explican a continuación.

4.5.3.1 Esquema de Limitación de nuevas llamadas (New Call Bounding Scheme)

En este esquema, se limita la admisión de *Nuevas Llamadas* dentro de las redes inalámbricas. El esquema opera como sigue: Si el número de *Llamadas Nuevas* en una celda excede un *valor de umbral* cuando llega una llamada, la *Nueva Llamada* será bloqueada; en otro caso, la llamada será admitida. Por otro lado, una *Llamada de Traspaso* es rechazada sólo cuando todos los canales en la celda están siendo utilizados.

Sea n_1 , que representa el *Número de Nuevas Llamadas* iniciadas en la celda; y sea n_2 , que representa el *Número de Llamadas de Traspaso* en la celda. Sea también K , el *Valor de Umbral* para las nuevas llamadas. Las expresiones para la *Probabilidad de*

Bloqueo de Nuevas Llamadas (P_{nb-nc}) y Probabilidad de Interrupción de llamadas de Traspaso (P_{hb-nc}) están dadas por,

Ecuación 4.74:

$$P_{nb-nc} = \frac{\sum_{n_2=0}^{C-K} \frac{\rho_n^K}{K!} * \frac{\rho_h^{n_2}}{n_2!} + \sum_{n_1=0}^{K-1} \frac{\rho_n^{n_1}}{n_1!} * \frac{\rho_h^{C-n_1}}{(C-n_1)!}}{\sum_{n_1=0}^K \frac{\rho_n^{n_1}}{n_1!} \sum_{n_2=0}^{C-n_1} \frac{\rho_h^{n_2}}{n_2!}}$$

Ecuación 4.75:

$$P_{hb-nc} = \frac{\sum_{n_1=0}^K \frac{\rho_n^{n_1}}{n_1!} * \frac{\rho_h^{C-n_1}}{(C-n_1)!}}{\sum_{n_1=0}^K \frac{\rho_n^{n_1}}{n_1!} \sum_{n_2=0}^{C-n_1} \frac{\rho_h^{n_2}}{n_2!}}$$

4.5.3.2 Esquema de Prioridad de Limitación (Cutoff Priority Scheme)

En este esquema, en lugar de poner limitación al *Número de Nuevas Llamadas*, se utiliza el *Número Total de Llamadas en Curso* en la celda para tomar una decisión acerca de aceptar o rechazar una *Nueva Llamada* entrante.

El esquema opera como sigue. Sea m , que representa el *Umbral* cuando ocurre la llegada de una *Nueva Llamada*. Si el *Número Total de Canales Ocupados* es menor que m , se aceptará la *Nueva Llamada*; de otra forma, la *Nueva Llamada* será rechazada. Las *Llamadas de Traspaso* son aceptadas siempre a menos que no haya un canal disponible cuando ocurre la llegada. Entonces, las expresiones para la *Probabilidad de Bloqueo de Nuevas Llamadas* y la *Probabilidad de Interrupción de Llamada de Traspaso* son,

Ecuación 4.76:

$$P_{nb-cop} = \frac{\sum_{j=m}^C \frac{(\rho_n + \rho_h)^m \rho_h^{j-m}}{j!}}{\sum_{j=0}^m \frac{(\rho_n + \rho_h)^j}{j!} + \sum_{j=m+1}^C \frac{(\rho_n + \rho_h)^m \rho_h^{j-m}}{j!}}$$

Ecuación 4.77:

$$P_{hb-cop} = \frac{\frac{(\rho_n + \rho_h)^m \rho_h^{C-m}}{C!}}{\sum_{j=0}^m \frac{(\rho_n + \rho_h)^j}{j!} + \sum_{j=m+1}^C \frac{(\rho_n + \rho_h)^m \rho_h^{j-m}}{j!}}$$

4.5.3.3 Número de flujos en el Encaminador de Acceso Inalámbrico

Ahora, se usarán las expresiones comprendidas entre la

Ecuación 4.74 y la

Ecuación 4.77, donde se obtuvieron P_{nb} y P_{hb} para ambos esquemas de CAC, para obtener el *Número Medio de Nuevas Llamadas* (N_n) y el *Número Medio de Llamadas de Traspaso* (N_h) para cada caso. Para ello, se utiliza la expresión para el *Número Medio de Elementos* en un sistema de pérdidas de Erlang dada por,

Ecuación 4.78:

$$N = \frac{\lambda(1 - P_b)}{\mu}$$

Entonces, se pueden obtener N_n y N_h de las expresiones de P_{nb} y P_{hb} respectivamente para cada modelo CAC así,

Ecuación 4.79:

$$N_n = \frac{\lambda_n(1 - P_{nb})}{\mu_n}$$

Ecuación 4.80:

$$N_h = \frac{\lambda_h(1 - P_{hb})}{\mu_h}$$

Por tanto, el número medio del total de llamadas en la celda para cada modelo CAC se calcula como,

Ecuación 4.81:

$$N_t = N_n + N_h$$

4.5.3.4 Retardo medio de los paquetes para el proceso de Clasificación y el proceso de Planificación

Para calcular el *Retardo Medio* de los paquetes en el proceso de *Clasificación* y el proceso de *Planificación*, se utilizan los mismos modelos obtenidos para estos procesos en la sección 4.3.

4.5.4 Resultados Obtenidos

Para propósitos de evaluación, se analizó solamente el *Retardo Medio* del proceso de *Clasificación* ya que es la principal diferencia entre ambas tecnologías (*IntServ* e *IntServ6*), mientras que el proceso de *Planificación* no representa una diferencia entre ellas.

Al igual que en la sección 4.3, para el análisis se utilizaron los parámetros del procesador de red *Intel IXP 1200*, que se observan en la

Tabla 4.3. Debido a que el retardo de los paquetes depende del *Número de Flujos* establecidos, y que a su vez el establecimiento de estos flujos es autorizado por el esquema de *Control de Admisión*, el comportamiento que se obtendrá es similar para *Encaminadores* con diferentes capacidades (en términos de *Número de Recursos* por asignar). El *Número de Recursos* por asignar varía dependiendo de la jerarquía del *Encaminador* en la *Red de Acceso*. En esta sección, a manera de ejemplo, se consideraron los valores típicos de una estación base de telefonía celular y que fueron los mismos utilizados en [105]. Por tanto, los análisis fueron realizados con un valor máximo de $C=30$ recursos por asignar en el *Encaminador de Acceso inalámbrico* y la *Tasa Media de Llegada de Ráfagas* es de 128 llegadas/s. Los valores de ρ_n y ρ_h fueron seleccionados de forma que se obtengan valores enteros del *Número de Recursos Asignados*, ya que en el entorno planteado, a cada flujo se asigna un valor fijo de recursos y sólo puede haber un número entero de flujos. También se tiene en cuenta que el *Número de Recursos Asignados* en una celda no puede ser mayor que el *Número de Recursos Asignados* por el esquema de CAC. Por tanto, el *Número de Flujos* no sobrepasará un límite que para el caso del esquema *New Call Bounding* será de 27 y para el caso del esquema *Cutoff Priority* será de 25.

La Figura 4.27 y la Figura 4.28 muestran los resultados de simulación para el comportamiento del *Retardo de las Ráfagas* de paquetes con respecto a la *Intensidad de Tráfico de Nuevas Llamadas* y *Llamadas de Traspaso* para los dos esquemas de CAC mencionados (*New Call Bounding* y *Cutoff Priority*) y las dos tecnologías de QoS (*IntServ* e *IntServ6*).

La Figura 4.27 muestra el comportamiento del *Retardo de las Ráfagas* de paquetes con respecto a la variación de la *Intensidad de Nuevas Llamadas*, manteniendo constante la *Intensidad de Tráfico de Llamadas de Traspaso* en un valor de 15. Por otro lado, la Figura 4.28 muestra el comportamiento del *Retardo de las Ráfagas* de paquetes con respecto a la variación de la *Intensidad de Llamadas de Traspaso*, manteniendo constante la *Intensidad de Tráfico* de nuevas llamadas en un valor de 15.

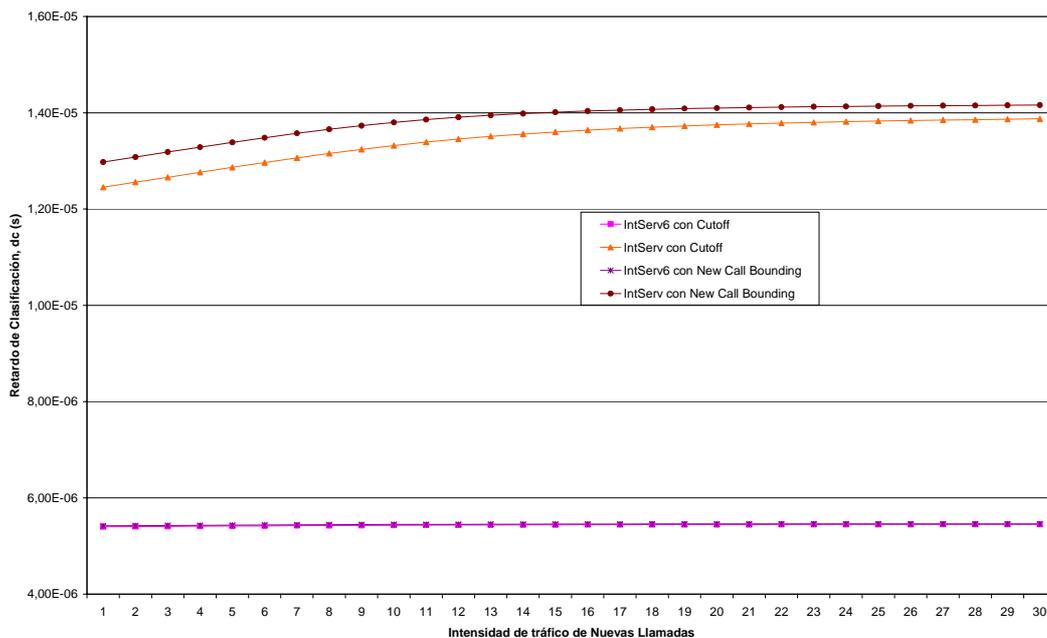


Figura 4.27 Variación del *Retardo Medio* de las ráfagas de paquetes para *IntServ* e *IntServ6*, bajo dos esquemas de *Control de Admisión*, con respecto a la *variación de la intensidad de tráfico de Nuevas Llamadas*.

Se puede observar en la Figura 4.27 y la Figura 4.28 que el *retardo medio de las ráfagas* para *IntServ* se incrementa cuando aumenta el *número de flujos*. Por otra parte, el *retardo medio de las ráfagas* para *IntServ6* se incrementa más lentamente, de manera

que incluso parece casi constante con respecto al comportamiento de *IntServ*. Cabe resaltar también que el *retardo de ráfaga* para *IntServ6* es de un orden de magnitud por debajo del mismo parámetro en *IntServ*.

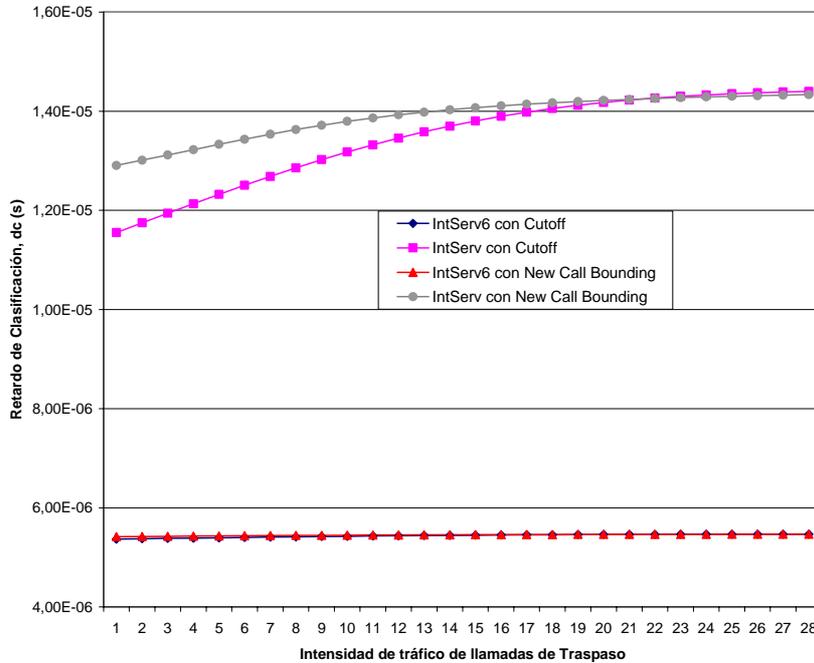


Figura 4.28 Variación del *Retardo Medio* de las ráfagas de paquetes para *IntServ* e *IntServ6*, bajo dos esquemas de *Control de Admisión*, con respecto a la variación de la *intensidad de tráfico* de llamadas de *Traspaso*.

Otra observación importante es que el esquema de *CAC* utilizado tiene un efecto sobre el comportamiento del *Retardo de las Ráfagas* en ambas tecnologías de *QoS*. Así, el esquema *Cutoff Priority Bounding* genera un menor retardo que el esquema *New Call Bounding*. De otro lado, ambos esquemas de *CAC* presentan una saturación del *Retardo de las Ráfagas*. Esto se debe a que se alcanza el *Número Máximo de Recursos Asignados*, por lo que el *Número Medio de Flujos* llega a su máximo límite y por tanto, se mantiene un *tiempo Medio de Clasificación* constante mientras se mantenga este *Número de Flujos*. Estos fenómenos son bastante notorios en el caso de *IntServ* mientras que para el caso de *IntServ6* son casi imperceptibles debido a que el *Tiempo de Búsqueda en la Tabla Hash* permanece constante debido al uso del *Tagging*, y el posible aumento de *Tiempo de Clasificación* sólo podría deberse al aumento en las

Colisiones, lo cual tiene un efecto menos notorio en la variación del *Retardo*. Por tanto, se puede concluir que *IntServ6* presenta una mayor independencia del retardo de clasificación con respecto a la movilidad de los usuarios, mientras que *IntServ* presenta una dependencia notoria con respecto al mismo fenómeno.

4.6 Evaluación del tiempo de Traspaso

Uno de los aspectos importantes a analizar en redes inalámbricas es el tiempo de duración (o *Latencia*) del *Traspaso*. Como es sabido, la interrupción en la comunicación durante este proceso debe ser mínima, por lo que se debe tratar de que el *Tiempo de Traspaso* sea lo más pequeño posible. Es así que en esta sección se realiza un análisis comparativo del *Tiempo de Traspaso* para las dos tecnologías de soporte de *Calidad de Servicio* a comparar: *IntServ* e *IntServ6*. Inicialmente se describe la arquitectura de red a utilizar para el análisis y posteriormente se realiza la comparación entre las dos tecnologías. Este análisis fue publicado en [12].

4.6.1 Arquitectura de red a utilizar

Para realizar la evaluación del *tiempo de Traspaso* partiremos de la arquitectura de red que se observa en la Figura 4.29. En esta red, se supone una situación común en que el enlace inalámbrico tiene una velocidad del orden de 11 Mbps y el enlace entre *ARs* y *MAPs* es de 100 Mbps, pues se supone que tanto el *MAP* como los *ARs* bajo su control están dentro de un mismo dominio de red que puede ser una red de área local o una red de área metropolitana. Finalmente, para la movilidad se utilizan los protocolos *HMIPv6* y *Fast Handoff*.

Para el estudio de tiempos de *Traspaso* no se tiene en cuenta el tiempo de propagación de la señal, pues esto depende de las distancias y medios de transmisión utilizados y complicaría mucho el modelo. Tampoco se tiene en cuenta el efecto de los errores en la transmisión. Únicamente se tiene en cuenta el efecto del *tiempo de transmisión* de los mensajes y el tiempo de procesamiento en los *Encaminadores*. Bajo estas consideraciones, se hace el análisis que se explica a continuación.

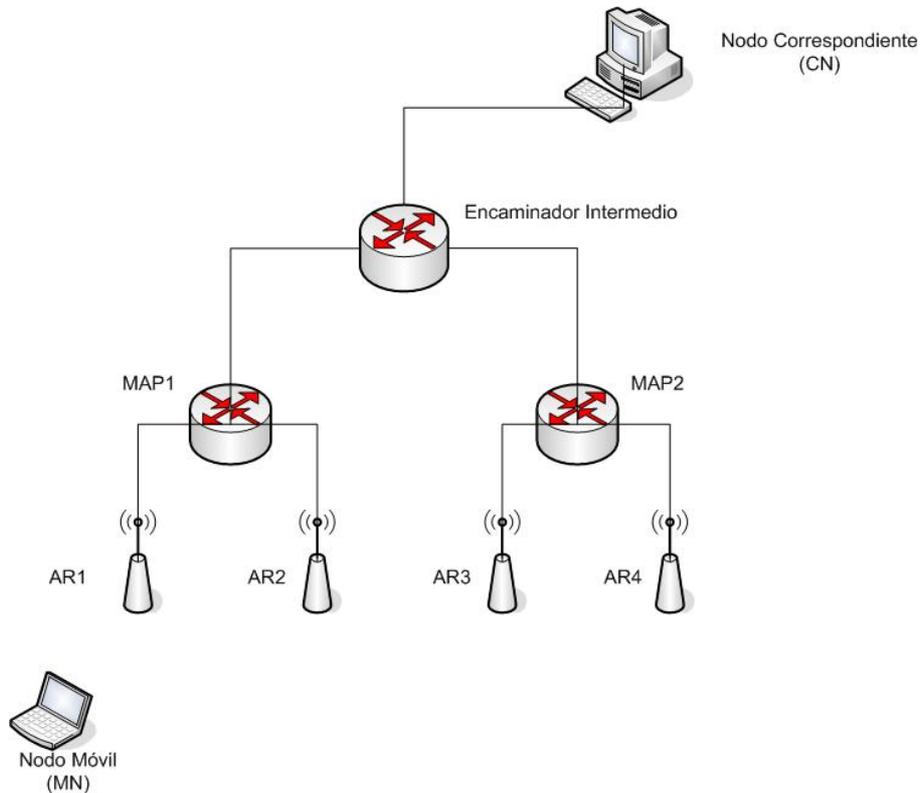


Figura 4.29 Arquitectura usada en HMIPv6.

4.6.2 Análisis del Ahorro en tiempo de Traspaso para IntServ6

Para la realización del *Traspaso*, tanto en la propuesta de *IntServ6* como en el estándar *IntServ*, se deben establecer reservas en el nuevo camino hacia el nodo móvil. La diferencia fundamental entre usar el estándar de *IntServ* e *IntServ6* es la forma en que se gestionan las *reservas en los túneles*. Para el caso de las reservas en túneles en *IntServ*, se utiliza el *RFC 2746*. Sin embargo, los pasos a seguir para establecer las reservas en los túneles son los mismos y los mensajes *PATH* y *RESV* intercambiados son similares. El intercambio de mensajes para el caso de *IntServ* se observa en la figuras 4.29 a 4.31. La figura Figura 4.30 muestra la fase de establecimiento de la reserva inicial, mientras que en la Figura 4.31 se observa el intercambio de mensajes durante el proceso de *Fast Handoff* (o *Fast Handover*) entre *ARs* de un mismo *MAP* (lo que se conoce como

Traspaso Local). Por último, la Figura 4.32 muestra la situación de *Traspaso* entre MAPs.

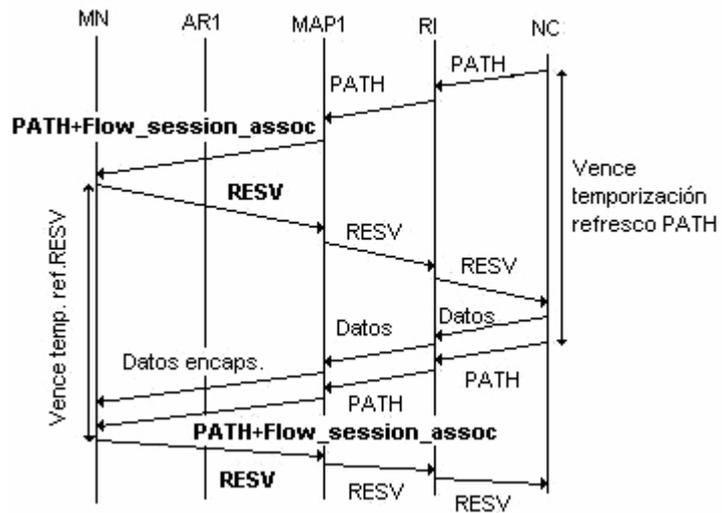


Figura 4.30 Fase de establecimiento de la reserva.

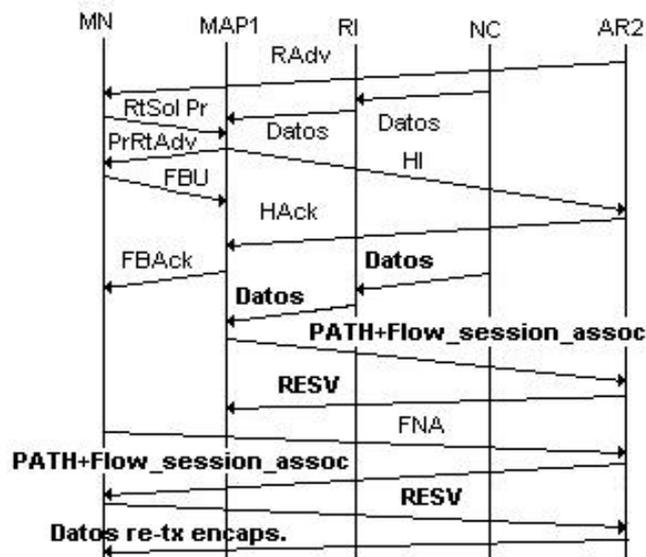


Figura 4.31 Fase de *Fast Handover* entre ARs de un mismo dominio MAP.

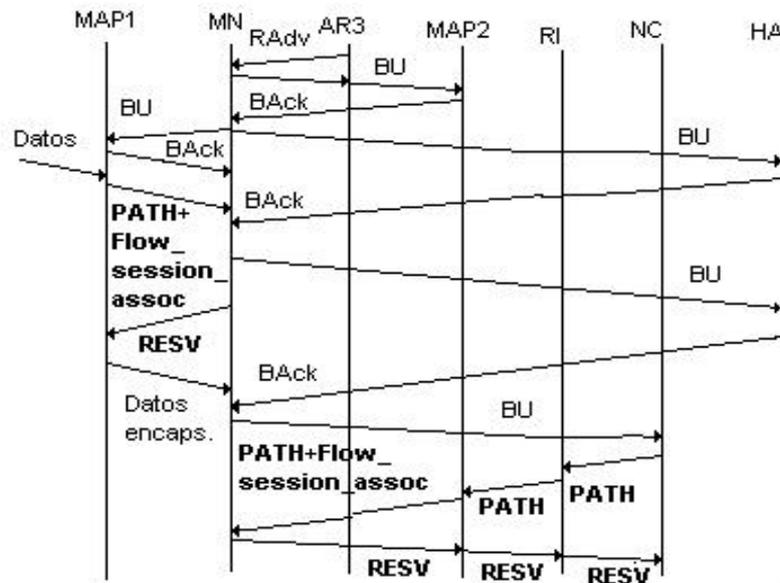


Figura 4.32 Fase de Traspaso entre MAPs.

El RFC 2746 que explica las reservas RSVP en los túneles describe que para establecerse la reserva en el túnel, debe enviarse un mensaje *PATH* junto con un objeto *SESSION_ASSOC*, que tiene 48 Bytes de longitud (incluyendo cabecera común, objeto *SESSION* y objeto *IPV6_FILTER_SPEC*). Este objeto se envía para relacionar las reservas *E2E* con las reservas internas del túnel, por lo que transporta los datos de la *Quíntupla* de la reserva *E2E* y de la reserva interna del túnel.

Para el caso de *IntServ6* el mensaje *PATH* no requiere ningún objeto adicional, pues se establece la reserva con la información que transporta el mensaje *PATH* normal pero cambiando el objeto *IPV6_SENDER_TEMPLATE* (*class=11, c-type=2*). En nuestra propuesta se deberá enviar este objeto pero del tipo *IPV6_FLOW_LABEL_SENDER_TEMPLATE* (*class=11, c-type=3*) para enviar el número hash que se transporta en la *Etiqueta de Flujo* y que identifica la reserva. Estos dos objetos tienen la misma longitud (20 Bytes), por lo que no cambia el número de Bytes enviados con respecto al establecimiento de una reserva común (que no se realiza sobre un túnel).

Por tanto, cada vez que se establezca una nueva reserva dentro de un túnel en el proceso de *Traspaso*, *IntServ6* ahorrará el tiempo necesario para transmitir los 48 Bytes utilizados para el objeto *SESSION_ASSOC* en el RFC 2746.

Además, de este tiempo de ahorro debido al menor número de *Bytes* transmitidos, *IntServ6* también ahorra el tiempo necesario para calcular el *número hash* en cada *Encaminador* del nuevo camino (incluyendo los *Encaminadores* del túnel) durante el establecimiento de la reserva, ya que este número es transmitido en el mismo mensaje *PATH*, tal como se mencionó anteriormente.

Por tanto, el ahorro en tiempo de *Traspaso* (T_s) para *IntServ6* puede calcularse como:

Ecuación 4.82:

$$T_s = \sum_i \left(\frac{1}{v_{L-i}} \cdot 48 \cdot 8 \right) + \sum_j T_{chash-j}$$

Es decir, el *tiempo ahorrado* es la sumatoria de los *tiempos de transmisión* de los 48 *Bytes* para cada enlace dentro del túnel (el tiempo de transmisión de 1 bit es el inverso de la velocidad de transmisión en el enlace i , v_{L-i}) en el nuevo camino más la sumatoria de los tiempos de cálculo del *número hash* (t_{chash}) en cada *Encaminador* del nuevo camino.

El tiempo de cálculo del *número hash* se puede aproximar si tenemos en cuenta los parámetros del procesador de red *IXP1200* como el tiempo necesario para leer la *Quíntupla* en palabras de 32 bits, más el tiempo necesario para escribir el resultado. Ya que la *Quíntupla* consta de 37 *Bytes*, es decir, 296 bits, se requerirán 10 lecturas de *SRAM*. Además, se requerirá una escritura en *SRAM* para escribir el resultado (*número hash* de 20 bits). Por tanto, el tiempo de cálculo del *número hash* será:

Ecuación 4.83:

$$T_{chash-j} = 11 L_{SRAM} = 11 \cdot 80ns = 880ns$$

Entonces, para la arquitectura de red de ejemplo, tenemos que para el caso de *Traspaso local*, se cambiará sólo en un enlace fijo de 100 Mbps y un enlace inalámbrico de 11Mbps, por lo que el ahorro en el tiempo de *Traspaso* con *IntServ6* respecto a *IntServ* será de,

Ecuación 4.84:

$$T_{sHOlocal} = ((1/10^8)*48*8) + ((1/(11*10^6))*48*8) + (2*880*10^{-9}) = 40,5 \mu s$$

Para el caso del *Traspaso* entre *MAPs*, se cambia el camino del túnel en un enlace de 100Mbps y un enlace inalámbrico de 11Mbps, por lo que el ahorro en el tiempo de *Traspaso* de *IntServ6* con respecto a *IntServ* será,

Ecuación 4.85:

$$T_{sHOMAP} = ((1/10^8)*48*8) + ((1/(11*10^6))*48*8) + (3*880*10^{-9}) = 41,38 \mu s$$

Si bien es cierto, el tiempo ahorrado con *IntServ6* en procesamiento es muy pequeño en comparación con los tiempos debidos al establecimiento de los caminos durante el *Traspaso*, está claro que *IntServ6* no aporta más retardo al *Traspaso*. En lugar de esto, la propuesta *IntServ6* presenta una pequeña mejoría en el *tiempo de Traspaso*, tanto para el caso de *Traspaso Local*, como para el caso de *Traspaso* entre *MAPs*. Esta mejora, aunque pequeña, representará una mejora considerable en el desempeño general de la red al aumentar el *número de Traspasos* y el *número de usuarios móviles* en la red.

4.7 Comparación de IntServ6 con MPLS en redes móviles basadas en IP

4.7.1 Introducción

El interés en usar *MPLS* como un esquema subyacente de transporte de *MIP* (arquitectura que se denominará en adelante *MIP/MPLS*) está justificado por la capacidad de *MPLS* de acomodarse a diferentes escalas y por las garantías de *QoS* que ofrece. También está la simplicidad asociada con el establecimiento y mantenimiento de las entidades de conmutación de etiquetas. Un aspecto adicional es la facilidad de integración con las extensiones de micro-movilidad y macro-movilidad y las extensiones de *QoS* que soportan movilidad. Otro factor que no puede ser pasado por alto es que *MPLS* puede ser una solución muy buena de inter-operatividad para *IPv4* e *IPv6*, ya sea a nivel cableado o inalámbrico. Todo esto, junto con la capacidad de

Ingeniería de Tráfico de *MPLS*, provee una solución relevante en el soporte de *QoS* para el usuario final móvil[106].

Sin embargo, un problema bastante relevante en las redes *MIP/MPLS* es la gestión de etiquetas, pues estas son un recurso finito de las redes *MPLS*, lo que hace que en *redes de acceso* con muchos usuarios, este recurso pueda llegar a agotarse en uno o varios segmentos de la red. Tal característica hace que, aunque *MPLS* se comporta muy bien en *redes de transporte*, presente problemas al ser usada como *red de acceso*. Por otro lado, *IntServ6* se comporta de mejor manera en el aspecto de gestión de etiquetas. Por tanto, en esta sección se realizará un análisis comparativo de las dos tecnologías.

4.7.2 Soluciones de MIP/MPLS

Bajo el entorno planteado, la existencia de usuarios móviles implica que los *LSPs* (*Label Switched Path*) establecidos requieren de movilidad pero la tecnología *MPLS* inicial fue diseñada inicialmente para usuarios fijos, un problema similar al que aparece en la Internet tradicional. En consecuencia, han surgido una serie de soluciones para este problema en que *MPLS* debe complementar al protocolo de movilidad *MIP*. Algunas de estas soluciones proponen que *MPLS* debe soportar dos tipos de *LSPs*: *LSPs* estáticos y *LSPs* dinámicos.

En [107], los autores sugieren la integración de las arquitecturas *MPLS* y *MIP*. Ellos describen primero una arquitectura donde el *FA* y el *HA* son *LSRs Frontera* y pertenecen al mismo dominio *MPLS*. Se describen también dos escenarios para los *FAs* y *HAs* que no son *LSRs Frontera* del mismo dominio *MPLS*. El primer escenario es cuando el dominio *MPLS local* y el dominio *MPLS Foráneo* están interconectados por una nube *MPLS*. El otro caso es cuando los dos dominios están interconectados a través de una nube *IP*. En este caso se sugiere que los túneles *MPLS* sean extendidos por túneles *IP* y se utiliza *HMIP*. Sin embargo, los autores notan que su esquema se desempeña mejor cuando la red completa es *MPLS*.

El trabajo en [108], llamado “*Mobility-Aware MPLS*”, propone usar *MPLS* al nivel de red de acceso, teniendo tanto dominios *MIP* como *MPLS* en la red de transporte (*Core*). Este asume un *Gateway (GW)* para el dominio a través del cual se accede a la red de transporte (*Core*). Se establecen dos tipos de *LSPs* entre las estaciones base (*BS*) y el *GW*. El primer tipo son *LSPs estáticos*; en este caso tanto el *down-link* como el *up-link* son *LSPs* pre-configurados por el administrador de red con el propósito de establecer *FECs* pre-definidos. Los *LSPs dinámicos*, por otro lado, son iniciados por el *MN* y son establecidos durante el registro entre el *MN* y el *GW* de la red. Si el *MN* está en una *red Foránea*, el registro se extiende hasta el *GW* de la *red local (Home Network)*. La *red de transporte*, con arquitectura *DiffServ/MPLS*, utiliza *LSPs estáticos*, mientras que la *red de acceso* utiliza los *LSPs dinámicos* que usan señalización *RSVP*.

Hierarchical Mobile MPLS [109] extiende la propuesta de integración *MIP/MPLS* descrita antes para el caso donde la red completa es *MPLS*. Esta propone una jerarquía doblemente enlazada, con un nuevo agente de movilidad llamado el *FDA (Foreign Domain Agent)*, en cada dominio *MPLS*. Además, para mejorar la entrega de reservas durante un *Traspaso*, el artículo toma prestada la idea de *reservas activas y pasivas* que ha sido propuesta varias veces en las extensiones de *RSVP* para *MIP* (p.ej. [110]). En efecto, los autores extendieron su trabajo en *HM-MPLS* en [111] para acomodar una solución *IntServ* sobre *DiffServ*. También, en la adaptación de su trabajo hacia ambientes pico-celulares, ellos utilizaron *RSVP-TE* como un protocolo de distribución de etiquetas, tomando ventaja de su característica de *Soft-State*.

La solución denominada “*MPLS-Based Hierarchical MIP*” en [112] es similar al *HMIP*, excepto que reemplaza la mensajería y el entunelamiento con *MPLS* hasta donde es viable. Esta propuesta difiere, sin embargo, por su definición de re-Encaminamiento *LSP*, que se refiere a la re-dirección del *LSP* debido a los *Traspasos*. Ella define tres categorías: 1) Re-establecimiento total de *LSP*, 2) Re-Encaminamiento dinámico basado en reservas adelantadas en redes vecinas, 3) Re-Encaminamiento basado en *Multicast*.

Hay otros dos estudios que vale la pena mencionar. Uno presentado en [113], detalla una arquitectura unificada para la gestión de la movilidad inter-dominio e intra-dominio, que brinda funcionalidades para optimización del Encaminamiento y el *paging*. Esta

última, extraída de la operación celular, es usada para la ubicación eficiente de los *MNs*. El segundo estudio [114] brinda una mirada dentro de la gestión de etiquetas, cuando en una solución integrada se usan tanto *LSPs* dinámicos como estáticos entre el *HA* y el *MN*. Esto propone que un *LSP* no debería ser establecido tan pronto como el *MN* se registra con su *HA* sino hasta que el *CN* haya iniciado su flujo. Esta propuesta también estudia la noción de la generación de actualizaciones de etiquetas en el *HA* si un *LSP* hacia un *MN* está ocioso más allá de un cierto umbral de tiempo.

En la recomendación *Y.1281* [115] de la *UIT-T* se realiza un intento de estandarización hacia este tema. Esta discute los detalles de arquitectura de implementar *MIP* sobre *MPLS*, tanto para *MIPv4* como para *MIPv6*. También discute la implementación para una solución *HMIP*. En los aspectos dignos de mención de la recomendación, está el énfasis en la capacidad de *MPLS* en proveer la optimización del Encaminamiento de capa 2 y también *MPLS* como una solución para la migración desde *IPv4* hasta *IPv6* que implica la gestión de la movilidad *IP*.

4.7.3 La gestión de Etiquetas en MIP/MPLS

Uno de los principales problemas que se establece en la recomendación *Y.1281* de la *UIT-T* [115] es que la gestión de etiquetas requiere una solución más elaborada, pues las etiquetas deben ser consideradas como un recurso de la red que debe ser gestionado apropiadamente.

Un trabajo relacionado con este tema es el realizado en [114]. Como se mencionó antes, esta propuesta brinda una mirada dentro de la gestión de etiquetas, cuando en una solución integrada se usan tanto *LSPs dinámicos* como *estáticos* entre el *HA* y el *MN*. Específicamente, dicho artículo se enfoca en la gestión de *LSPs dinámicos*, ya que los *LSPs estáticos* son gestionados mediante los protocolos tradicionales de Encaminamiento y distribución de etiquetas (*LDP*, *Label Distribution Protocol*).

Para realizar la gestión de los *LSPs* dinámicos, en [114] se propone la actualización de etiquetas de dichos *LSPs* y para obtener tal fin se consideran varios aspectos:

➤ *Asignación de Etiquetas*

Una etiqueta es equivalente a un camino desde el punto de vista de un *LSR* (*Label Switched Router*). Estas etiquetas son un recurso limitado de la red, no solo por el número de posibles combinaciones con una longitud de etiqueta dada, sino por aspectos prácticos al momento de la implementación. Además, aparecen problemas en la gestión del ancho de banda para el soporte de *QoS* cuando aumenta el número de etiquetas.

➤ *Establecimiento de LSPs dinámicos*

Para el establecimiento de los *LSP* dinámicos debe haber un registro previo a nivel de *MIP*. En esta aproximación, el nuevo *LSP* se establece cuando ya hay una cierta cantidad de tráfico fluyendo y, para efectos prácticos, se considera un umbral de tráfico en el *HA*.

➤ *Eliminación de LSPs dinámicos*

Igualmente, si un *LSP* dinámico no presenta tráfico durante un cierto umbral de tiempo, se desechará. Esto es necesario, pues el *LSR* no puede ser notificado para que elimine el flujo debido a la naturaleza sin conexión de *Internet*. Los dos umbrales establecidos para el establecimiento y la eliminación de los *LSPs dinámicos* determinarán el tiempo de vida de los *LSPs* y por tanto afectarán el número de etiquetas utilizadas en cada momento, el porcentaje de tráfico transmitido mediante *MPLS* y la tasa de pérdida de paquetes cuando se mueve el *MN*.

4.7.4 Desempeño de la gestión de etiquetas en MIP/MPLS

Aunque la propuesta planteada en [114] sólo presenta resultados de simulaciones y no plantea algunos modelos matemáticos que describan las relaciones planteadas, es claro que estudia ciertos aspectos interesantes y obtiene también resultados importantes que se describen a continuación.

➤ *Condiciones de simulación*

Se consideran varios tipos de paquetes entre los que se encuentran: Los enviados mediante *MIP* solamente debido al *Traspaso* (denominados paquetes de almacenamiento y re-envío), los paquetes perdidos y los enviados mediante *MPLS* (*cut-through packets*).

Se simula una red *MPLS* (ver Figura 4.33) con varios *LSRs* que comunican a los *HA* y *FA* con el *Núcleo* de la red *MPLS*. Existe un solo *HA* y siete *FA*. Además, existe un camino único para llegar hasta el *HA* a través de 3 *LSRs* consecutivos.

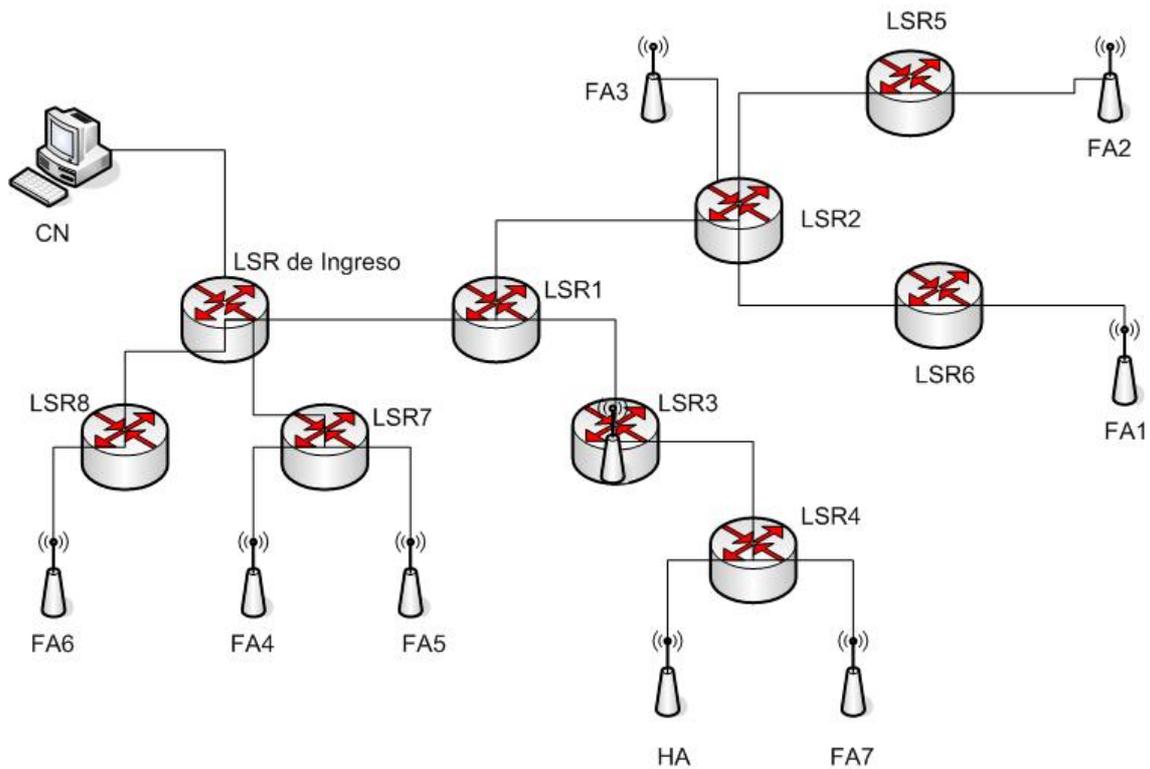


Figura 4.33. Arquitectura de la red simulada

➤ *Número de paquetes vs. Número de etiquetas dinámicas*

En este tipo de simulaciones se mantiene constante el número de nodos móviles en 500 *MNs* y se varía el número de etiquetas dinámicas en los *LSRs*. Como resultado (ver Figura 4.34), el porcentaje de paquetes enviados mediante *MPLS* se incrementa en la

medida que se incrementa el número de etiquetas en cada *LSR*. Esto es lógico, pues se aumenta el número de recursos de la red *MPLS* y por tanto se aumenta la capacidad de transporte de paquetes.

Adicionalmente, se obtiene que el número de paquetes transportado por la red *MPLS* se incrementa drásticamente hasta el 100 por ciento cuando se alcanza un número de etiquetas dinámicas que es el doble del número de móviles. Esto se debe a que sólo hay un *HA* y se utiliza una etiqueta para cada sentido de la comunicación entre el *HA* y el *MN*, por lo que en realidad se alcanza un número de recursos igual al número de usuarios (1 recurso = 2 etiquetas).

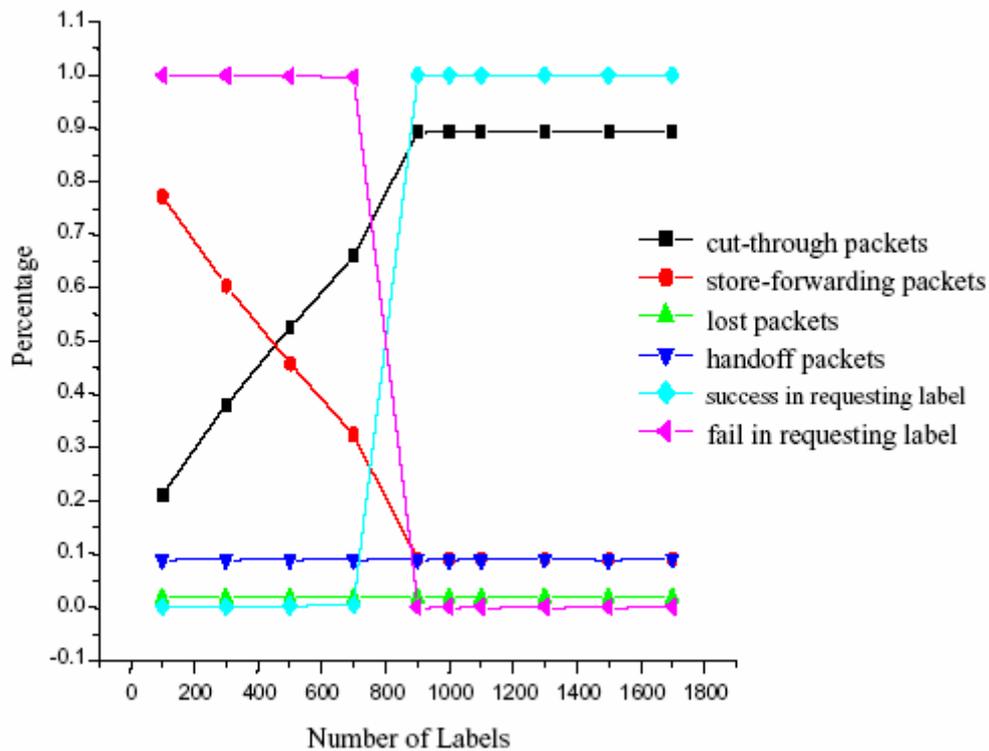


Figura 4.34 Porcentaje de paquetes entregados con respecto al número de etiquetas dinámicas.[114]

➤ *Número de paquetes vs. Número de Nodos Móviles*

En este tipo de simulaciones (ver Figura 4.35) se mantiene constante el número de etiquetas en 1000 etiquetas, mientras que se varía el número de Nodos Móviles. Como

resultado, el porcentaje de paquetes transportados por *MPLS* decae dramáticamente cuando se obtiene un número de usuarios mayor que la mitad de las etiquetas dinámicas.

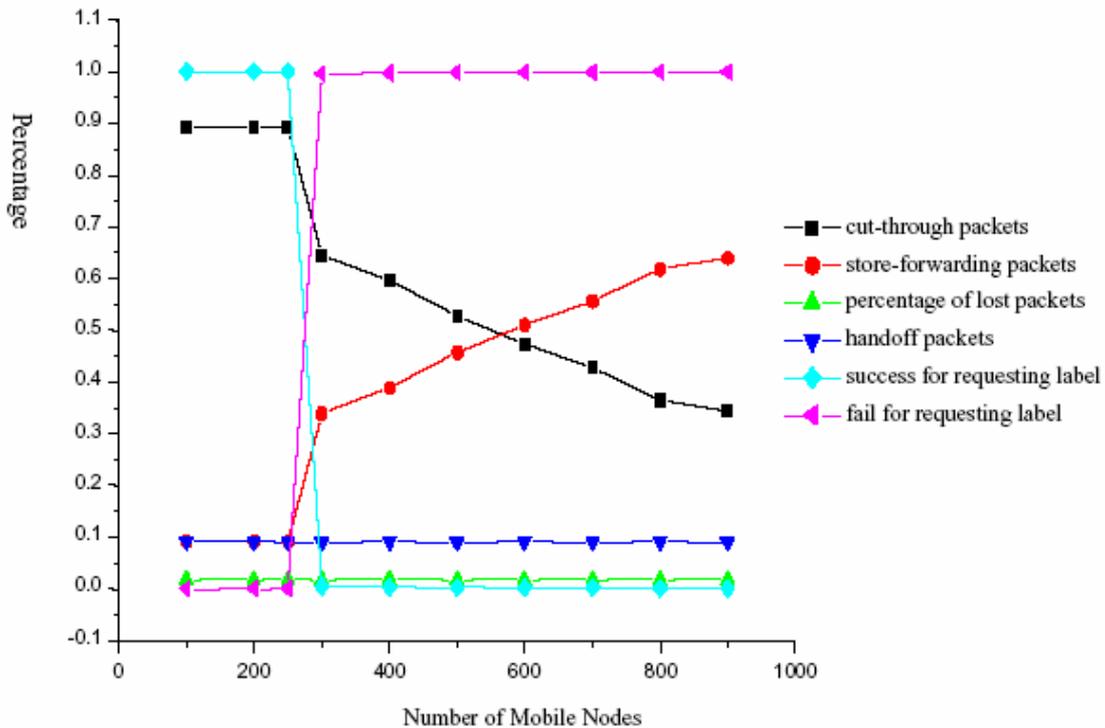


Figura 4.35 Porcentaje de paquetes entregados con respecto al número de Nodos Móviles[114]

La razón es la misma explicada para el caso anterior, es decir, debido a que cada usuario móvil utiliza dos etiquetas (una para cada sentido de la comunicación) en el *LSR* frontera que comunica al *HA*, al sobrepasar un número de usuarios que utilizan más de las etiquetas existentes, se produce un bloqueo de los usuarios nuevos entrantes y el porcentaje de los paquetes transportados mediante *MPLS* con respecto al número total de paquetes generados por los usuarios móviles decae drásticamente.

➤ *Consumo de etiquetas en cada LSR*

En estas simulaciones se tuvieron dos situaciones: La primera que mantiene constante el número de móviles (500) y varía el número de etiquetas, y la segunda que mantiene constante el número de etiquetas (1000) y varía el número de móviles. Para ambas situaciones se graficó el número de etiquetas sin usar en cada *LSR*. Se puede observar en la Figura 4.36 que el número de etiquetas no usadas en los *LSR* 1, 3 y 4 es casi nulo

pues los paquetes de todos los m viles deben transitar hacia/desde el *HA* a trav s de estos *LSRs*. S lo empieza a haber etiquetas sin usar en estos *LSRs* cuando se supera el l mite de dos etiquetas por cada m vil.

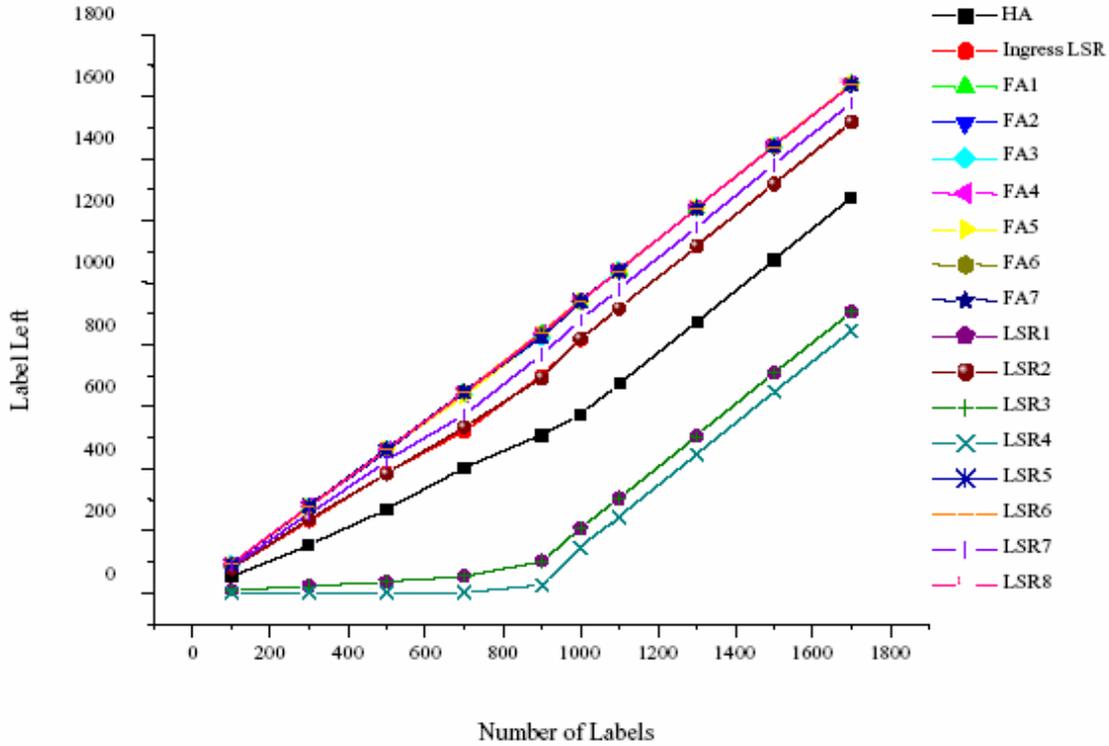


Figura 4.36 Consumo de etiquetas en cada *LSR* respecto al n mero de etiquetas asignadas [114].

Por otro lado, cuando se mantiene constante el n mero de etiquetas (1000) y se var a el n mero de m viles, se obtienen los resultados de la Figura 4.37 para el n mero de etiquetas no usadas en cada *LSR*. Puede observarse que en el *LSR 4* (que interconecta al *HA*) el n mero de etiquetas sin usar va decreciendo hasta alcanzar un valor de cero cuando se sobrepasa un n mero de m viles tal que consume el total de etiquetas del *LSR 4* (la mitad del n mero de etiquetas disponibles).

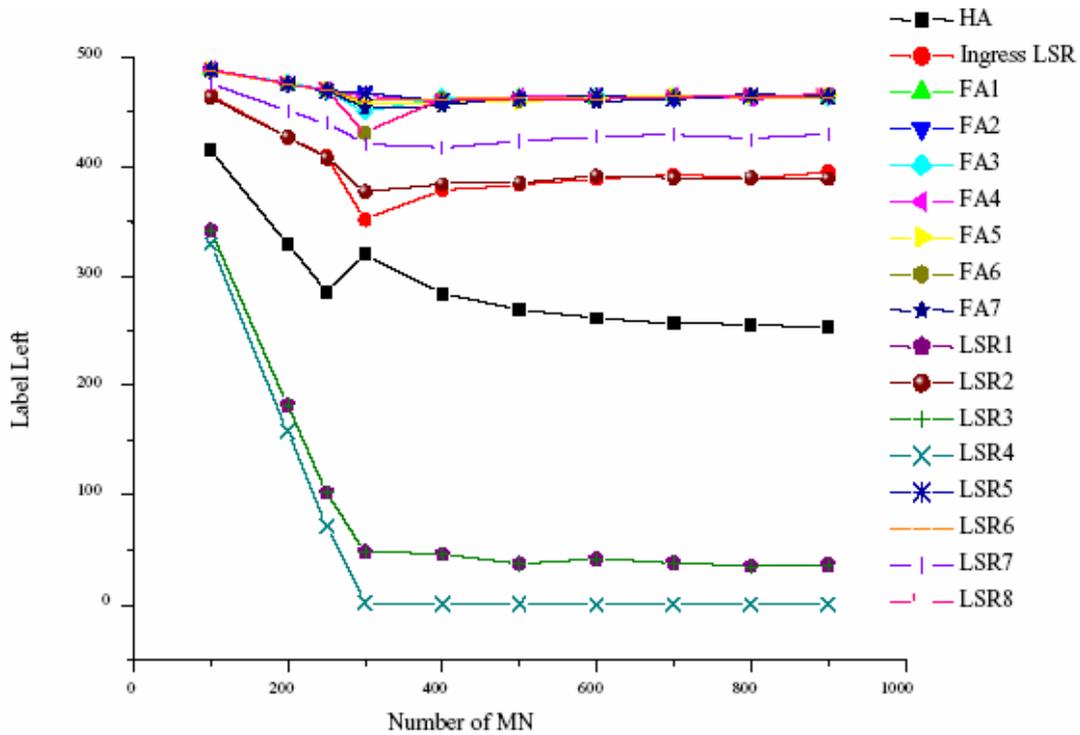


Figura 4.37 Consumo de etiquetas en cada *LSR* respecto al número de móviles [114].

Finalmente, en [114] se obtiene una tabla que muestra el porcentaje de paquetes transportados por *MPLS* respecto al número de etiquetas en los *LSRs* 1 y 4, manteniendo constante el número de *MNs* en 500. Al graficar esta tabla se obtiene la curva de la Figura 4.38.

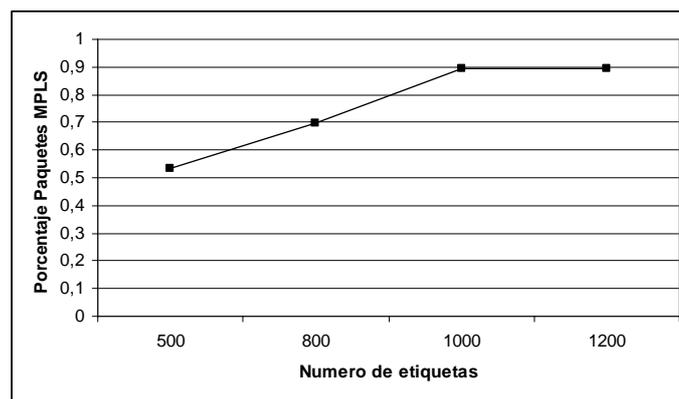


Figura 4.38 Porcentaje de paquetes transportados por *MPLS* respecto al número de etiquetas.

Puede observarse en la Figura 4.38 que el número de paquetes transportados por *MPLS* puede incrementarse si se aumenta el número de etiquetas. Sin embargo, el porcentaje

de paquetes transportados se satura cuando el número de etiquetas excede el número de etiquetas necesarias para comunicar los *MNs* existentes con el *HA* en ambos sentidos (*upstream* y *downstream*).

4.7.5 Comparación de MPLS con IntServ6 en redes de acceso MIP

Es claro que en redes *MPLS* las etiquetas son un recurso finito que hace que el número de usuarios que puede servir la red *MPLS* llegue a un punto de bloqueo. Este bloqueo por consumo de etiquetas no ocurre en *IntServ6* pues aunque el número de claves *Hash* es de 2^{20} , la función *Hash* siempre asignará una etiqueta a cualquier comunicación (ver sección 3.2). Por otro lado para *IntServ6*, el aumentar el número de usuarios hace que se aumente el número de colisiones, de manera que pueden seguir aumentando los usuarios pero el número de colisiones podría hacer que se llegue a retardos muy grandes e intolerables para algunos tipos de servicios, tal como se observó en las secciones 4.1 y 4.2.

Por otra parte, aunque en *IntServ6* no se llega a un bloqueo por el consumo de etiquetas, sí se llega a un bloqueo por la cantidad de recursos que se han asignado en un *Encaminador* determinado. Es claro que el módulo de *CAC* de los *Encaminadores* es el encargado de determinar si los recursos son suficientes para asignarlos a una reserva que se está solicitando y dependiendo del método empleado por el esquema de *CAC*, se obtendrá una probabilidad de bloqueo. Esto pudo observarse en la sección 4.3.

En cuanto a la escalabilidad, como se mencionó en las secciones 3.3 y 3.4, *IntServ6* utiliza un procedimiento similar al *Label Stacking* de *MPLS* para manejar las reservas dentro de túneles y la agregación de flujos, lo que hace que *IntServ6* solucione el problema de escalabilidad que presenta el estándar *IntServ*. Es decir, en este sentido *IntServ6* se comporta de forma similar a *MPLS*.

4.8 Conclusiones

En este capítulo se describieron diferentes modelos matemáticos y de simulación que permiten evaluar el desempeño de la propuesta *IntServ6* con respecto al estándar *IntServ*; en algunos aspectos *IntServ6* se comparó también con *MPLS*. Los modelos matemáticos obtenidos fueron contrastados con los modelos de simulación y se realizaron los ajustes necesarios para obtener un comportamiento cercano al de las simulaciones. De otra parte, los modelos de colas del proceso de *Clasificación* se obtuvieron bajo condiciones de tráfico *Semi-Elastico* que genera llegadas de ráfagas de paquetes. Según las pruebas estadísticas realizadas, el modelo de colas se ajusta mejor para cantidades de flujos en un rango entre 70 y 1200 aproximadamente. Todos los aspectos evaluados arrojaron que definitivamente *IntServ6* es superior a *IntServ*. Así, *IntServ6* mejora el rendimiento de los *Encaminadores* en aspectos como el *Ancho de Banda de Memoria* utilizado y la velocidad de procesamiento; *IntServ6* también mejora el *Tiempo Medio de Clasificación* y el *Tiempo Medio de Retardo* de los paquetes, además mejora el *Tráfico Servido* a cada flujo en *Reservas Simples* y en las *Reservas en Túneles*. Por otro lado, *IntServ6* reduce considerablemente la dependencia del retardo de los paquetes con respecto a la movilidad de los usuarios y además afecta de forma mínima e incluso reduce el *Tiempo de Latencia de Traspaso*.

Finalmente, si se compara *IntServ6* con *MPLS* en entornos de redes de acceso móviles basadas en *IP* se puede concluir que:

- Es claro que en redes *MPLS* las etiquetas son un recurso finito que hace que el número de usuarios que puede servir la red *MPLS* llegue a un punto de bloqueo.
- Por otro lado para *IntServ6*, el aumentar el número de usuarios hace que se aumente el número de colisiones, de manera que pueden seguir aumentando los usuarios (si el esquema de control de acceso lo permite) pero el número de colisiones podría hacer que se llegue a retardos muy grandes e intolerables para algunos tipos de servicios.

- Aunque en *IntServ6* no se llega a un bloqueo por el consumo de etiquetas, sí se llega a un bloqueo por la cantidad de recursos que se han asignado en un *Encaminador* determinado.

- En cuanto a la escalabilidad, *IntServ6* utiliza un procedimiento similar al *Label Stacking* de *MPLS* para manejar las reservas dentro de túneles y la agregación de flujos, lo que hace que *IntServ6* solucione el problema de escalabilidad que presenta el estándar *IntServ*. Es decir, en este sentido *IntServ6* se comporta de forma similar a *MPLS*.

5 Conclusiones y Líneas Futuras

En esta Tesis se plantea una solución para soportar *Calidad de Servicio* en *Internet*. Esta solución, denominada *IntServ6*, opera sobre redes fijas y móviles basadas en el protocolo *IPv6*. *IntServ6* está fundamentada en la *Arquitectura de Servicios Integrados* y utiliza el campo de *Etiqueta de Flujo* de la cabecera *IPv6* para identificar los flujos que tienen reservas de recursos en los *Encaminadores*. La utilización del campo *Etiqueta de Flujo* de *IPv6* no ha sido estandarizada, existiendo hasta el momento sólo ciertas especificaciones de uso y algunas propuestas de su utilización que también fueron estudiadas. Tales propuestas sólo abarcan algunos aspectos de una posible solución pero no plantean soluciones completas. Estos aspectos son entre otros, el formato de interpretación de la *Etiqueta de Flujo* y algunas definiciones de arquitecturas de *QoS* descritas de forma muy general.

Una característica importante de *IntServ6* es que ofrece soluciones simples y eficientes a problemas tales como el *Soporte de Reservas dentro de Túneles* y la *Agregación de Flujos*. Para ambos procesos, *IntServ6* utiliza un procedimiento similar al *Label Stacking* de *MPLS*. Debido a que *IntServ6* permite la *Agregación de Flujos* de una forma natural, mejora las características de escalabilidad que presenta la solución *IntServ* estándar. *IntServ6* también logra solucionar el problema de *Reservas dentro de Túneles* a nivel de *capa 3 (Internet)* mientras que la solución estándar de *IntServ* debe acudir a información de la *capa 4 (Transporte)*. Esto conlleva a una mayor rapidez en el tratamiento de los paquetes en los *Encaminadores*, mejorando la cantidad de tráfico servido en ellos.

Adicionalmente, *IntServ6* posee extensiones para operar sobre redes móviles. Tales extensiones están diseñadas para operar con los protocolos de movilidad existentes de forma que no requieren la alteración de dichos protocolos. Aunque *IntServ6* podría operar con el protocolo *Mobile IPv6*, es preferible su operación sobre *HMIPv6* para reducir el tiempo de *Traspaso*.

En cuanto a la evaluación de *IntServ6*, se obtuvieron diferentes modelos matemáticos y de simulación que permitieron evaluar el rendimiento de la propuesta *IntServ6* con respecto al estándar *IntServ* y, en algunos aspectos, se comparó también con *MPLS*. Todos los aspectos evaluados arrojaron que definitivamente *IntServ6* es superior a *IntServ*. Así, *IntServ6* mejora el rendimiento de los *Encaminadores* en aspectos como el *Ancho de Banda de Memoria* y la velocidad de procesamiento, también mejora el *Tiempo Medio de Retardo* de los paquetes, mejora el *Tráfico Servido* a cada flujo en *Reservas Simples* y en las *Reservas en Túneles*. Por otro lado, *IntServ6* reduce considerablemente la dependencia del *Retardo* de los paquetes con respecto a la movilidad de los usuarios y además no afecta e incluso reduce un poco el *Tiempo de Latencia de Traspaso*.

Si se hace una comparación entre *MPLS* e *IntServ6*, *MPLS* es una solución que requiere de *Encaminadores* especiales que realizan la *Conmutación de Etiquetas*, de forma que para agilizar el Encaminamiento de los paquetes *IP*, se requiere de una red de transporte *MPLS* que lleve los paquetes desde una red *IP* origen hasta una red *IP* destino. La ventaja que supone esta solución, además de que es un medio de transporte rápido, es que funciona tanto con paquetes *IPv4* como con paquetes *IPv6*, a la vez que permite hacer Encaminamiento con restricciones de *Calidad de Servicio*. Por su parte, *IntServ6* puede ser parte de la arquitectura de un Encaminador de Internet y por tanto, no requiere de Encaminadores especiales, aunque de otro lado, *IntServ6* opera específicamente para *IPv6*.

Adicionalmente, si se compara *IntServ6* con *MPLS* en entornos de *Redes de Acceso Móviles* basadas en *IP*, se puede concluir que en *MPLS* las *Etiquetas* son un recurso finito que hace que el *Número de Flujos* que puede servir la red *MPLS* llegue a un punto de bloqueo. Este bloqueo por consumo de *Etiquetas* no ocurre en *IntServ6* pues aunque el número de claves *Hash* es sólo de 2^{20} , la función *Hash* siempre asignará una *Clave Hash* a cualquier comunicación que sea aceptada por el módulo de *Control de Admisión*. Por otro lado para *IntServ6*, el aumentar el *Número de Flujos* hace que se aumente el *Número de Colisiones*, de manera que pueden seguir aumentando los *Flujos* pero el *Número de Colisiones* podría hacer que se llegue a retardos muy grandes e intolerables para algunos tipos de servicios. Por otro lado, aunque en *IntServ6* no se

llega a un bloqueo por el consumo de *Etiquetas*, sí se llega a un bloqueo por la cantidad de recursos que se han asignado en un *Encaminador* determinado. Por tanto, dependiendo del método empleado por el esquema de CAC, se obtendrá una probabilidad de bloqueo diferente.

Como todo trabajo de investigación, no se puede pretender que el tema tratado en esta Tesis esté completamente terminado y cerrado. Por el contrario, aparecen una serie de expectativas que serán objeto de trabajos posteriores que continuarán con la investigación descrita en este documento. Tales temas a desarrollar son los siguientes:

- Al modelo actual de los *Encaminadores de Servicios Integrados* puede agregarse el efecto del *control de Políticas*.
- Un aspecto que es importante es el método para establecer el total de recursos que serán reservados para un *Agregado de flujos*. Este tema queda abierto en la propuesta actual. En principio, se puede usar la sumatoria de los recursos de todos los flujos que conforman el *Agregado*, pero esta opción podría hacer un uso ineficiente de los recursos de los *Encaminadores*.
- Hasta el momento, se han hecho estudios en ciertas situaciones que permiten un tratamiento matemático mediante herramientas como la teoría de colas con colas M/M/1 o colas M/G/1. Este tipo de situaciones comprenden servicios de tráfico semi-elástico tales como el servicio de transferencia de ficheros (FTP) en los que los datos llegan a ráfagas y se puede asumir que estas ráfagas tienen llegadas con distribución exponencial. Sin embargo, es necesario atacar otras situaciones tales como el tráfico típico de una Internet en que se puede asumir un comportamiento auto-similar. Se debe tratar de simular una fuente auto-similar por métodos como el de multiplexación de fuentes, modelos MMPP u otros, que permitan generar llegadas de paquetes a un *Encaminador* de Servicios Integrados basado en IntServ6 para determinar el comportamiento bajo estas condiciones.

- Un tema de gran actualidad es el de las redes de sensores. Existen algunos estudios acerca de *Calidad de Servicio* en este tipo de redes y se han propuesto algunos parámetros de *QoS* particulares para estas redes. Podrían hacerse adaptaciones a *IntServ6* y estudiar este tipo de parámetros para así analizar el comportamiento de *IntServ6* este tipo de entornos.
- Una tecnología que tiene una gran aceptación hoy en día es el de la distribución de canales de televisión por *Internet*. Uno de los trabajos posibles es el de hacer estudios del comportamiento de los *Encaminadores IntServ6* con este tipo de tráfico.
- El paso siguiente en el desarrollo de una nueva tecnología después de hacer la evaluación matemática y de simulaciones es la fase de construcción. Por lo que un posible trabajo es la construcción de un *Encaminador IntServ6 fijo* y también un *Encaminador IntServ6* Inalámbrico. Finalmente, se podría construir un prototipo de una *Red de Acceso* basada en *IP Móvil* o *HMIP* para hacer pruebas que incluyan aspectos de movilidad de los usuarios.
- Otro aspecto que ocupa una buena cantidad de investigadores hoy en día es el *Encaminamiento con Calidad de Servicio*. Este se ha estudiado hasta el momento con *MPLS* (ya se han hecho algunas publicaciones en este sentido [13-15]). Sin embargo, podrían hacerse modificaciones a la propuesta actual de *IntServ6* para ofrecer este tipo de complementos y permitir utilizar la *Etiqueta de Flujo* de la cabecera *IPv6* para hacer este tipo de Encaminamiento.

6 Lista de Acrónimos

AR	Access Router
ATM	Asynchronous Transfer Mode
Back	Binding Acknowledge
BB	Bandwidth Broker
BU	Binding Update
BW	Bandwidth
CIDR	Classless Inter Domain Routing
CN	Correspondent Node
CoA	Care of Address
CRC	Cyclic Redundancy Check
CR-LDP	Constraint Based Routing Label Distribution Protocol
DCoA	Domain Care of Address
DiffServ	Differentiated Services Architecture
DS	Differentiated Services Field
DSCP	Differentiated Services Code Point
EBB	Exponentially Bounded Burstiness
E2E	End-to-End
FA	Foreign Agent
FEC	Forward Equivalente Class
FIFO	First-In First-Out
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GPS	Generalized Processor Sharing
HA	Home Agent
HMIP	Hierarchical Mobile Internet Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services Standard
IntServ6	Servicios Integrados para IPv6
IP	Internet Protocol
LCoA	Local Care of Address

LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switching Path
LSR	Label Switching Router
MAP	Mobility Anchor Point
MIP	Mobile Internet Protocol
MIP-LR	Mobile IP with Location Registers
MN	Mobile Node
MPLS	Multi-Protocol Label Switching
PGPS	Packet-by-Packet GPS
PHB	Per-hop Behavior
QoS	Quality of Service
RSVP	Resource Reservation Protocol
RA_{dv}	Router Advertisement
RCoA	Regional Care of Address
SLA	Service Level Agreement
SLS	Service Level Specification
TC	Traffic Class
TCP	Transport Control Protocol
TOS	Type of Service
UDP	User Datagram Protocol
UIT	Unión Internacional de Telecomunicaciones
UMTS	Universal Mobile Telecommunications System
VLSM	Variable-Length Sub-net Mask
VPN	Virtual Private Network
WFQ	Weighted Fair Queuing

7 Publicaciones relacionadas con la Tesis Doctoral

A continuación se detallan las publicaciones producto de esta Tesis Doctoral:

Revistas Internacionales Indexadas

- Padilla, J. and Paradells, J. 2007. IntServ6: An approach to support QoS over IPv6 wired and wireless networks. *European Transactions on Telecommunications*. ISSN 1124-318X. 2007. (Publicado en Línea: Junio 14 de 2007).

Congresos Internacionales

- Padilla, J., Paradells, J. and Rodríguez, A. “Supporting QoS over IPv6 wireless networks with IntServ6”. *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06)*. INSPEC Accession Number:9164978. 2006.
- Padilla, J., Paradells, J. and Rodríguez, A. “Effects of Mobility on Mean Packet Delay over Integrated Services Wireless Networks”. *2nd IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'06)*. ISBN 1-4244-0517-3. 2006.
- Padilla J., Paradells, J., Huerta, M. and Hesselbach, X. “IntServ6: An Approach to Support QoS over IPv6 Networks”. *10th IEEE Symposium on Computers and Communications Proceedings (ISCC'05)*. ISSN: 1530-1346. 2005.

- Huerta, M., Padilla J., Hesselbach X, Fabregat R. and Ravelo O. “Buffer Capacity Allocation: A method to QoS support on MPLS networks”. *IEEE EuroAmerican Conference on Telematics and Information Systems (EATIS)*. ISBN 958-8166-38-1. 2006.
- Padilla, J., Paradells, J., Huerta, M. and Hesselbach, X. “QoS Support over IPv6 Networks”. *IEEE INFOCOM Student Workshop*. 2005.
- Padilla J., Paradells J. “Una propuesta para soportar Calidad del servicio en redes móviles”. *IEEE, 2do. Congreso Internacional de la Región Andina, Andescon*. ISBN 958-33-6534-3. 2004.

Revistas Internacionales

- Huerta Mónica, Hesselbach X., Fabregat R., Padilla J., Ravelo O. and Clotet R. “Reducción de Congestión mediante técnicas de optimización de flujos en redes MPLS”. *Latin America Transactions*. ISSN 1548-0992. 2007. Aceptado para publicación.

Revistas Nacionales

- Padilla J., Paradells J., Huerta M., Hesselbach X. “Soporte de QoS sobre IPv6 con IntServ6”. *Gerencia y Tecnología Informática*. Vol. 4. Num. 8. ISSN 1657-8236. Abril de 2005.
- Huerta, M., Padilla J., Hesselbach X, Fabregat R. and Ravelo O. “Buffer Capacity Allocation: A method to QoS support on MPLS networks”. *Colombian Journal of Computation*. ISSN 1657-2831. Junio de 2006.

8 Bibliografía

- [1] C. Perkins, "IP Mobility Support for IPv4," IETF, RFC 3344 2002.
- [2] C. Perkins, "Mobility Support in IPv6," IETF, RFC 3775 2004.
- [3] H. Soliman y C. Castelluccia, *Hierarchical Mobile IPv6 mobility Management (HMIPv6)*: IETF RFC 4140, 2005.
- [4] A. T. Campbell, J. Gomez, S. Kim, A. G. Valko, C.-Y. Wan, y Z. R. Turanyi, "Design, implementation, and evaluation of cellular IP", *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 7, pp. 42-49, 2000.
- [5] A. T. Campbell, J. Gomez, S. Kim, C.-Y. Wan, Z. R. Turanyi, y A. G. Valko, "Comparison of IP micromobility protocols", *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 9, pp. 72-82, 2002.
- [6] J. Padilla y J. Paradells, "IntServ6: An approach to support QoS over IPv6 wired and wireless networks", *European Transactions on Telecommunications*, 2007.
- [7] J. Padilla, J. Paradells, M. Huerta, y X. Hesselbach, "QoS Support over IPv6 Networks," IEEE INFOCOM Student Workshop, 2005.
- [8] J. Padilla, M. Huerta, J. Paradells, y X. Hesselbach, "IntServ6: An Approach to Support QoS over IPv6 Networks," 10th IEEE Symposium on Computers and Communications Proceedings., 2005.
- [9] J. Padilla, J. Paradells, H. M., y H. X., "Soporte de QoS sobre IPv6 con IntServ6", *Gerencia y Tecnología Informática*, vol. 4, pp. 11-20, 2005.
- [10] J. Padilla, J. Paradells, y A. Rodríguez, "Supporting QoS over IPv6 wireless networks with IntServ6," The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'06), 2006.
- [11] J. Padilla, J. Paradells, y A. Rodríguez, "Effects of Mobility on Mean Packet Delay over Integrated Services Wireless Networks," 2nd IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'06), 2006.
- [12] J. Padilla y J. Paradells, "Una propuesta para soportar Calidad del servicio en redes móviles," 2do. Congreso IEEE de la Region Andina. Andescon, 2004.
- [13] M. Huerta, J. Padilla, X. Hesselbach, R. Fabregat, y O. Ravelo, "Buffer Capacity Allocation: A method to QoS support on MPLS networks," IEEE EuroAmerican Conference on Telematics and Information Systems, 2006.
- [14] M. Huerta, J. Padilla, R. Fabregat, X. Hesselbach, y O. Ravelo, "Buffer Capacity Allocation: A method to QoS support on MPLS networks", *Colombian Journal of Computation*, 2006.
- [15] M. Huerta, H. Hesselbach, R. Fabregat, J. Padilla, O. Ravelo, y R. Clotet, "Reducción de Congestión mediante técnicas de optimización de flujos en redes MPLS", *Latin America Transactions*, 2007 (Aceptado para publicación).
- [16] C. Huitema, *IPv6: The New Internet Protocol*, 2da ed. New Jersey: Prentice Hall, 1998.
- [17] S. Hagen, *IPv6 Essentials*, 2da ed: O'reilly Media Inc., 2006.
- [18] S. Deering, *Internet Protocol version 6 (IPV6)*: IETF RFC 2460, 1998.
- [19] R. Braden, *Integrated Services in the Internet Architecture: an Overview*: IETF RFC 1633, 1994.

- [20] S. Blake, D. Black, y M. Carlson, *An Architecture for Differentiated Services*: IETF RFC 2475, 1998.
- [21] B. Jamoussi, *Multiprotocol Label Switching Architecture*: IETF RFC 3031, 1999.
- [22] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, 1st edition ed: Morgan Kaufmann, 2001.
- [23] R. Braden y L. Zhang, *Resource Reservation Protocol (RSVP)*: IETF RFC 2205, 1997.
- [24] W. Stallings, *Comunicaciones y redes de computadores*, 6 ed: Prentice Hall, 2000.
- [25] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, y E. Felstaine., *A Framework for Integrated Services Operation over Diffserv Networks*: IETF RFC 2998, 2000.
- [26] A. Abella, V. Friderikos, y H. Aghvami, "Differentiated services versus over-provisioned best-effort for pure-IP mobile networks," 4th International Workshop on Mobile and Wireless Communications Network., 2002.
- [27] Y. Cheng y W. Zhuang, "DiffServ resource allocation for fast handoff in wireless mobile Internet", *Communications Magazine, IEEE*, vol. 40, pp. 130-136, 2002.
- [28] B. Jamoussi, L. Andersson, y R. Callon, *Constraint-Based LSP Setup using LDP*: IETF RFC 3212, 2002.
- [29] J. Ash, *Applicability Statement for CR-LDP*: IETF RFC 3037, 2002.
- [30] A. Farrel, D. Papadimitriou, J.-P. Vasseur, y A. Ayyangar, "Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using RSVP-TE," IETF, draft-ietf-mpls-rsvpte-attributes-04.txt 2004.
- [31] J. L. Marzo, E. Calle, C. Scoglio, y T. Anjali, "Adding QoS protection in order to enhance MPLS QoS routing," IEEE International Conference on Communications, 2003. ICC '03., 2003.
- [32] L. Zhang, S. Deering, D. Estrin, S. Shenker, y D. Zappala, "RSVP: a new resource ReSerVation Protocol", *Network, IEEE*, vol. 7, pp. 8-18, 1993.
- [33] J. Kaur, *Scalable Network architectures for providing per-flow Service Guarantees*, Tesis Doctoral, University of Texas, Austin, 2002
- [34] B. Hardekopf, T. Riché, J. Mudigonda, M. Dahlin, y H. Vin, "Impact of Network Protocols on Programmable Router Architectures," Laboratory of Advanced systems Research, Department of Computer Science, University of Texas, Austin 2003.
- [35] J. Kaur, *Scalable Network architectures for providing per-flow service guarantees*, Faculty of the Graduate School, Dissertation. The University of Texas at Austin, Austin, Texas, 2002
- [36] A. K. Parekh y R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344-357, 1993.
- [37] C. E. Perkins, "Mobile IP", *Communications Magazine, IEEE*, vol. 35, pp. 84-99, 1997.
- [38] C. E. Perkins, "Mobile IP", *Communications Magazine, IEEE*, vol. 40, pp. 66-82, 2002.
- [39] J. Kempf, *Goals for Network-Based Localized Mobility Management (NELMM)*: IETF RFC 4831, 2007.

-
- [40] C. Perkins, *IP Encapsulation within IP*: IETF RFC 2003, 1996.
 - [41] C. Perkins, *Minimal Encapsulation within IP*: IETF RFC 2004, 1996.
 - [42] A. Conta y S. Deering, "Generic Packet Tunneling in IPv6," IETF RFC 2473 1998.
 - [43] A. K. Talukdar, B. R. Badrinath, y A. Acharya, "MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts", *Wireless Networks*, vol. 7, pp. 5-19, 2001.
 - [44] H. Chaskar, *Requirements of a Quality of Service (QoS) Solution for Mobile IP*: IETF RFC 3583, 2003.
 - [45] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, y L. Stibor, "IEEE 802.11e Wireless LAN for Quality of Service," *European Wireless*, 2002.
 - [46] C. Andreoli, N. Blefari-Melazzi, M. Listanti, y M. Palermo, "Mobility management in IP networks providing real-time services," 5th IEEE International Conference on Universal Personal Communications, 1996. Record., 1996.
 - [47] D. O. Awduche y E. Agu, "Mobile extensions to RSVP," *Computer Communications and Networks, 1997. Proceedings., Sixth International Conference on*, 1997.
 - [48] R. Jain, T. Raleigh, C. Graff, y M. Bereschinsky, "Mobile Internet access and QoS guarantees using mobile IP and RSVP with location registers," *IEEE International Conference on Communications, 1998. ICC 98. Conference Record.*, 1998.
 - [49] J. Indulska y S. Cook, "New extension to Support Computer Mobility in the Internet", *IEEE SICON*, 1998.
 - [50] C. C. Foo y K. C. Chua, "Implementing resource reservations for mobile hosts in the Internet using RSVP and mobile IP," *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, 2000.
 - [51] A. Terzis, *RSVP Operation Over IP Tunnels*: IETF RFC 2746, 2000.
 - [52] S. Paskalis, A. Kaloxylos, E. Zervas, y L. Merakos, "An Efficient RSVP-Mobile IP Interworking Scheme", *Mobile Networks and Applications*, vol. 8, pp. 197-207, 2003.
 - [53] I. Mahadevan y K. Sivalingam, "Quality of Service in Wireless Networks Based on Differentiated Services Architecture," *8th International Conference on Computer Communications and Networks*, 1999.
 - [54] I. Mahadevan y K. M. Sivalingam, "Architecture and Experimental Framework for Supporting QoS in Wireless Networks Using Differentiated Services", *Mobile Networks and Applications*, vol. 6, pp. 385-395, 2001.
 - [55] T. Braun y G. Stattenberger, "Providing Differentiated Services to Mobile IP Users," *26th Annual IEEE Conference on Local Computer Networks*, 2001.
 - [56] G. Stattenberger y T. Braun, "QoS Provisioning for Mobile IP Users," *1st International Workshop on Services and Applications in the Wireless Public Infrastructure*, 2001.
 - [57] P. Ford, Y. Rekhter, y H. Braun, "Improving the Routing and Addressing of IP", *IEEE Network*, pp. 10-15, 1993.
 - [58] Y. Rekhter, *CIDR and Classful Routing*: IETF RFC 1817, 1995.
 - [59] V. Fuller, T. Li, J. Yu, y K. Varadhan, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*: IETF RFC 1519, 1993.
 - [60] C. Partridge, *Using the Flow Label Field in IPv6*: IETF RFC 1809, 1995.
 - [61] A. Conta y S. Deering, *IPv6 Flow Label Specification*: IETF RFC 3697, 2004.

- [62] F. Baker y Y. Rekhter, "The use of Flow label for Tag Switching," IETF, Internet draft 1996.
- [63] Y. Rekhter, B. Davie, D. Katz, E. Rosen, y G. Swallow, *Cisco Systems' Tag Switching Architecture Overview*: IETF RFC 2105, 1997.
- [64] V. Roesler, L. Balbinot, M. de Andrade, y L. Tarouco, "IP next generation label switching," IP Operations and Management, 2002 IEEE Workshop on, 2002.
- [65] F. Ben Slimane y G. Pujolle, "Signaling in IP cell-switching," IEEE International Symposium on Computers and Communications, 1999. Proceedings., 1999.
- [66] X. Tang, J. Tang, G.-B. Huang, y C.-K. Siew, "QoS provisioning using IPv6 flow label in the internet," Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia., 2003.
- [67] E.-B. Fgee, W. J. Phillips, W. Robertson, y S. C. Sivakumar, "Implementing QoS capabilities in IPv6 networks and comparison with MPLS and RSVP," Canadian Conference on Electrical and Computer Engineering, 2003. IEEE CCECE 2003., 2003.
- [68] E. Fgee, J. D. Kenney, W. J. Phillips, W. Robertson, y S. Sivakumar, "Comparison of QoS performance between IPv6 QoS management model and IntServ and DiffServ QoS models," Proceedings of the 3rd Annual Communication Networks and Services Research Conference (CNSR'05), 2005.
- [69] F. Baker y C. Iturralde, *Aggregation of RSVP for IPv4 and IPv6 Reservations*: IETF RFC 3175, 2001.
- [70] F. L. Faucheur, *Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels*: IETF RFC 4804, 2007.
- [71] R. Koodli, *Fast Handovers for Mobile Ipv6*: IETF RFC 4068, 2005.
- [72] C. Bernardos, I. Soto, J. Moreno, T. Melia, M. Liebsch, y R. Schmitz, "Experimental evaluation of a Handover optimization solution for multimedia applications in a mobile IPv6 network", *European Transactions on Telecommunications*, vol. 16, pp. 317-328, 2004.
- [73] L. Galindo y P. Ruiz, "QoS and Micromobility Coupling: Improving Performance in Integrated Scenarios", *UPGRADE*, vol. VI, pp. 49-55, 2005.
- [74] G. Memik y W. Mangione-Smith, "Evaluating Network Processors Using NetBench", *ACM Transactions on Embedded Computing Systems*, vol. 5, pp. 453-471, 2006.
- [75] T. Hou y W. Lai, "Implementation of Network Processor-Based DiffServ/MPLS Edge router." Publicado: Intel, 2006.
- [76] CISCO, *CISCO Contents Services Gateway Data Sheet*: CISCO, 2006.
- [77] T. Spalink, S. Karlin, L. Peterson, y Y. Gottlieb, "Building a Robust SoftwareBased Router Using Network Processors," 18th SOSP, 2001.
- [78] M. Burns, G. Prier, J. Mirkovic, y P. Reiher, "Implementing Address Assurance in the Intel IXP Router." Publicado, 2002.
- [79] A. Sangiovanni-Vincentelli y A. Pinto, "An Overview of Embedded System Design Education at Berkeley", *ACM Transactions on Embedded Computing Systems*, vol. 4, 2005.
- [80] T. Wolf, "Network Systems Lab. Web Page. <http://www.ecs.umass.edu/ece/wolf/nsl/>." Publicado: University of Massachusetts., 2006.
- [81] Intel, "Post Workshop Report," Intel® IXA Workshop 2005, 2005.

- [82] I. Papaefstathiou, T. Orphanoudakis, G. Kornaros, C. Kachris, I. Mavroidis, y A. Nikolgiannis, "Queue Management in Network Processors," Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05), 2005.
- [83] Intel, *Intel® IXP1200 Network Processor Family. Hardware Reference Manual*: Intel, 2001.
- [84] D. Awduche, L. Berger, y D. Gan, *RSVP-TE: Extensions to RSVP for LSP Tunnels*: IETF RFC 3209, 2001.
- [85] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, 1st edition ed: Morgan Kaufmann, March 15, 2001.
- [86] J. Zhang, T. Hamalainen, y J. Joutsensalo, "Stochastic analysis of upper delay bound of GPS-based packetized fair queueing algorithms", *12th IEEE International Conference on Networks, 2004. (ICON 2004). Proceedings.*, vol. 1, pp. 35-40, 2004.
- [87] T. Bonald y J. W. Roberts, "Congestion at flow level and the impact of user behaviour", *Computer Networks*, vol. 42, pp. 521-536, 2003.
- [88] A. K. Parekh y R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case," *IEEE INFOCOM'93*, 1993.
- [89] O. Yaron y M. Sidi, "Generalized Processor Sharing Networks with Exponentially Bounded Burstiness Arrivals", *High Speed Network*, pp. 628-634, 1994.
- [90] O. Yaron y M. Sidi, "Performance and Stability of Communication Networks via Robust Exponential Bounds", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 1993, 1993.
- [91] Z. Zhang, D. Towsley, y J. Kurose, "Statistical Analysis of Generalized Processor Sharing Sheduling Discipline," *SIGCOMM'94*, 1994.
- [92] N. Chee Hock, *Queueing modeling fundamentals*: John Wiley & sons Ltd., 1997.
- [93] A. Feldmann, "Characteristics of TCP connection arrivals," en *Self-Similar Network Traffic and Performance Evaluation*: Wiley, 2000.
- [94] V. Paxson y S. Floyd, "Wide-area Taffic: The Failure of Poisson Modeling", *IEEE/ACM Transactions on Networking*, vol. 3, pp. 226-255, 1995.
- [95] T. Bonald, A. Proutiere, G. Régnié, y J. Roberts, "Insensitivity Results in Statistical Bandwidth Sharing," *ITC17 Teletraffic Engineering in the Internet Age*, 2001.
- [96] M. Crovella y A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Cause," *ACM SIGMETRICS'96*, 1996.
- [97] J. Pazos, A. Suárez, y R. Díaz, *Teoría de Colas y Simulación de Eventos Discretos*. Madrid: Pearson Educación S.A., 2003.
- [98] A. Hall, *Generalized Method of Moments*. New York: Oxford University Press, 2005.
- [99] D. Hong y S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures", *Vehicular Technology, IEEE Transactions on*, vol. 49, pp. 2037-2039, 2000.
- [100] R. Ramjee, R. Nagarajan, y D. Towsley, "On optimal call admission control in cellular networks," *INFOCOM '96. Fifteenth Annual Joint Conference of the*

- IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE, 1996.
- [101] S. Wu, K. Wong, y B. Li, "A dynamic call admission policy with precision QoS guarantee using stochastic control for mobile wireless networks", *IEEE/ACM Transaction on Networking*, vol. 10, pp. 257-271, 2002.
- [102] F. Yu, L. Ma, y V. Leung, "Effects of Mobility on Packet Level Performance of Connection-Oriented Multimedia Traffic over Packet-Switched Cellular Networks", *IEEE WCNC*, pp. 2171-2176, 2004.
- [103] Y. Fang, I. Chlamtac, y Y. Lin, "Teletraffic analysis and mobility modeling for PCS networks", *IEEE Transactions on Communications*, vol. 47, pp. 1062-1072, 1999.
- [104] Y. Fang, I. Chlamtac, y Y. Lin, "Channel occupancy times and handoff rate for mobile computing and PCS networks", *IEEE transactions on Comput.*, vol. 47, pp. 679-692, 1998.
- [105] Y. Fang y Y. Zhang, "Call admission control schemes and performance analysis in wireless mobile networks", *IEEE Transactions on Vehicular Technology*, vol. 51, pp. 371-382, 2002.
- [106] A. Taha, H. Hassanein, y H. Mouftah, "Integrated Solutions for Wireless MPLS and Mobile IP: Current Status and Future Directions," CCECE- CCGEI, 2004.
- [107] Z. Ren, C. Tham, C. Foo, y C. Koo, "Integration of Mobile IP and Multi-Protocol Label Switching," IEEE International Conference on Communications, 2001.
- [108] H. Kim, K. Wong, W. Chen, y C. Lau, "Mobility-Aware MPLS in IP-Based Wireless Access Networks," IEEE Global Telecommunications Conference, 2001.
- [109] T. Yang y D. Makrakis, "Hierarchical Mobile MPLS: Supporting Delay Sensitive Applications Over Wireless Internet," International Conference on Info-Tech and Info-Net, 2001.
- [110] I. Mahadevan y K. Sivalingam, "An Experimental Architecture for Providing QoS Guarantees in Mobile Networks using RSVP," 9th IEEE International Symposium on Personal, Indoor Mobile Radio Communications, 1998.
- [111] T. Yang, Y. Dong, Y. Zhang, y D. Makrakis, "Mobility Support for Differentiated Services in Next Generation MPLS-based Wireless Networks," International Conference on Communications, Circuits and Systems, and West Sino Exposition, 2002.
- [112] T. Um y J. Choi, "A Study on Path Re-Routing Algorithms at the MPLS Based Hierarchical Mobile IP Network," IEEE Region 10 International Conference on Electrical and Electronic Technology, 2001.
- [113] K. Xie y V. Leung, "A MPLS Framework for Macro- and Micro-Mobility Management," 2nd IASTED International Conference on Wireless and Optical Communications, 2002.
- [114] Y. Chen y Z. Yan, "Effect of the Label Management in Mobile IP over MPLS Networks," 17th International Conference on Advanced Information Networking and Applications (AINA'03), 2003.
- [115] *Recommendation Y.1281 Mobile IP Services Over MPLS*: ITU-T, 2003.