UNIVERSIDAD NACIONAL DEL CALLAO

ESCUELA DE POSGRADO

SECCIÓN DE POSGRADO DE LA FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

"IMPLEMENTACIÓN DE CALIDAD DE SERVICIO VÍA MULTIPROTOCOL LABEL SWITCHING (MPLS)"

TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS DE LA ELECTRÓNICA
MENCIÓN: TELECOMUNICACIONES

AUTOR: JORGE GUSTAVO BUTLER BLACKER

CALLAO – PERU
2011

Id. Ejemplar: 39210

JURADO SUSTENTACIÓN DE TESIS DE MAESTRÍA

DR. JUAN HERBER GRADOS GAMARRA	Presidente
MG. NICANOR RAÚL BENITES SARVIA	Secretario
DR. MARCELO NEMESIO DAMAS NIÑO	Miembro
MG. FRANCO IVÁN VÉLIZ LIZÁRRAGA	Miembro
MG. MANUEL ARIAS BARANDIARAN	Asesor

Nº DE LIBRO : 01

Nº DE ACTA : 01

FECHA: Marzo 23, 2011

A Dios que siempre me brindo el apoyo requerido en todos los momentos.
A mi familia por su apoyo.

Mi agradecimiento a:
Mg. Manuel Arias Barandarián y a la Ing. Miriam Oliveros Oliveros por su apoyo incondicional para el desarrollo del presente trabajo de investigación.

INDICE

PRÓLOGO VI

RESUMEN VII

ABSTRACT VIII

CAPÍTULO I 1

PLANTEAMIENTO INICIAL DE LA INVESTIGACIÓN 1

1.1 Identificación del problema 1

1.2 Formulación del problema 1

1.3 Objetivos de la investigación 2

1.4 Justificación 2

1.5 Limitaciones y facilidades 4

1.6 Hipótesis de partida 4

CAPITULO II 6

MARCO TEÓRICO 6

2.1 Antecedentes del estúdio 6

2.2 Definición de términos 8

CAPITULO III 13

METODOLOGÍA 13

3.1 Relación entre las variables de la investigación 13

3.2 Tipo de investigación 13

3.3 Diseño de la investigación 13

3.3.1 Calidad de servicio en la Internet 13

a) Proyecto de un modelo de QoS 14

- Ámbito de Aplicación 14
- Modelo de Control 14
- Garantía de Transmisión 15

b) Requisitos de QoS en la Internet2 15

- Aplicaciones avanzadas 16
- Interconectividad 20

- Interoperabilidad de equipos 21
- Escalabilidad 21
- Administrabilidad 21
- Mensurabilidad 22
- Requisitos de los Hosts 22
- Implementación incremental 23

3.3.2Técnicas para implementación de QoS 23

a) Servicios Integrados – RFC 1633 24

- Funcionamiento básico del protocolo RSVP 26

b) Servicios Diferenciados – RFC 2475 27

- Terminología 33
- Los componentes de una estructura de servicios diferenciados (DiffServ) 35
- Escalabilidad del servicio DiffServ 36

c) Un ejemplo de implementación de servicio "express" y garantizado con DiffServ" 37

- Clasificación de paquetes 37
- Control, marcado y "shaping" en la entrada 38
- Gestión de colas y "scheduling" de paquetes 38

d) MPLS 41

- Los componentes de un dominio MPLS 43
- Actividades generadas en MPLS 44

e) Estructura de las etiquetas 47

f) Label Distribution Protocol 49

g) Operación del MPLS 52

- Módulo de control 52
- Módulo de enrutamiento 53

h) MPLS e Ingeniería de Tráfico (TE) 54

- Constrain Route 54
- El protocolo CR-LDP 55
- El formato del mensaje "Solicitación de etiqueta" 57
- El formato del campo traffic TLV del mensaje "Solicitación de etiqueta" 58
- El formato del mensaje "Asignado de etiqueta" 61

i) Aplicaciones de MPLS 61
• Redes Privadas Virtuales (VPNs) 61
• Ingeniería de Tráfico (TE) 62
• Clase de servicio (CoS) 63
j) Un ejemplo de aplicación de MPLS 63
• Enrutamiento Convencional 63
• Enrutamiento con MPLS 64
k) Servicio DiffServ con MPLS 65
• DiffServ con MPLS asignado al campo EXP (E-LSP) 67
• DiffServ con MPLS asignado a etiqueta y campo EXP (L-LSP) 69
3.3.3 Evaluación de rendimiento de técnicas de implementación de QoS 70
a) Parámetros representativos de QoS 71
• Tráfico transportado (Throught) 71
• Retraso (Delay) 72
• Variación de retraso (Jitter) 72
• Perdida de paquetes (Packet lost) 73
b) Experiencias efectuadas 73
• Experiencia 1: IP convencional 74
• Experiencia 2: MPLS básico 75
• Experiencia 3: Caminos Explícitos.(Explicit LSPs) 75
• Experiencia 4: Clase de servicio (MPLS+CoS) 78
• Reserva de ancho de banda 78
• Prioridad IP 79
c) Equipos utilizados 80
• Microcomputadoras 80
• Routers 80
• Instrumentos de medición 80
• El software Netperf 81
d) Mediciones realizadas 83
• Mediciones de tráfico transportado 83
• Mediciones de retraso 83
• Topología para pruebas de retraso 83

3.4 Operacionalización de variables 84
3.5 Población y muestra 84
3.6 Técnicas e instrumentos de recolección de datos 84
3.7 Procedimientos de recolección de datos 84
3.8 Procesamiento estadístico y análisis de datos 84

CAPITULO IV 85

RESULTADOS 85

4.1 Experiencia 1: IP convencional (uno y dos flujos simultâneos) 85
4.1.1 Topologia 85
4.1.2 Configuración de los Routers 85
4.1.3 Procedimiento de La prueba 85
4.1.4 Dos flujos simultáneos en el enlace de 2 Mbps 85
4.1.5 Gráficos resultantes 86
4.2 Experiencia 2: MPLS básico (uno y dos flujos) 87
4.2.1 Topologia 87
4.2.2 Configuración de los Routers 87
4.2.3 Procedimiento de La prueba 88
4.2.4 Gráficos resultantes 88
4.3 Experiencia 3A: Caminos explícitos (uno y dos flujos) 89
4.3.1 Topología 89
4.3.2 Configuración de los routers 89
4.3.3 Procedimiento de la prueba parte A 89
4.3.4 Procedimiento de la prueba parte B 89
4.3.5 Gráficos resultantes 90
4.4 Experiencia 3B: Dos caminos explícitos y dos flujos diversificados 91
4.4.1 Topologia 91
4.4.2 Configuración de los routers 91
4.4.3 Procedimiento de la prueba 91
4.4.4 Gráficos resultantes 91
4.5 Experiencia 4A: CoS y reserva de ancho de banda 92
4.5.1 Topología 92
4.5.2 Configuración de los routers 92

4.5.3 Procedimiento de la prueba 92
4.5.4 Gráficos resultantes 93
4.6 Experiencia 4B: CoS, prioridad IP 94
4.6.1 Topología 94
4.6.2 Configuración de los routers 94
4.6.3 Procedimiento de la prueba B1 94
4.6.4 Procedimiento de la prueba B2 94
4.6.5 Gráficos resultantes 94
CAPITULO V 103
DISCUSION DE RESULTADOS 103
5.1 Implementación de QoS con MPLS 103
5.1.1 Asignación de recursos 105
5.1.2 Escalabilidad 106
5.1.3 Congestión 106
5.1.4 Continuidad de servicio 106
5.2 Parte experimental 106
CONCLUSIONES 110
RECOMENDACIONES 111
REFERENCIALES 112
ANEXOS 118
Matriz de consistencia 118

PROLOGO

La presente tesis de Maestría en Ciencias de la Electrónica con mención en Telecomunicaciones forma parte de los requerimientos académicos para la obtención del titulo de grado de Maestro que otorga la Facultad de Ingeniería Eléctrica y Electrónica (FIEE) de la Universidad Nacional del Callao (UNAC), según consta en los planes de estudio vigentes aprobados por la RESOLUCION RECTORAL Nº 1359-09-R DE 29 DE DICIEMBRE DE 2009.

Como asesor de esta tesis fue elegido el profesor Manuel Arias Barandiaran, quien propuso el tema de tesis, el cual fue remitido a la Comisión responsable de la Facultad, donde fue evaluado y aprobado según consta en la Resolución Directoral de la Sección Postgrado de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional del Callao Nº 004-2011-DSPG-FIEE con fecha 02 de Febrero del 2011.

Este trabajo se ha realizado con ayuda del Simulador OPNET IT GURU versión Académica, instalado en el Laboratorio de Informática de la FIEE y parte en las instalaciones del autor

RESUMEN

Debido la expansión del fenómeno *Internet* y, consecuentemente, del fenómeno **IP**, han surgido nuevas formas de tráfico que desafían las premisas básicas del proyecto de redes **IP**. Estos tráficos, flujos multimedia en tiempo real, no solamente demandan grandes tasas de transmisión sino también imponen exigencias particulares de *timing* como baja tolerancia a retraso y altas garantías fin-a-fin de entrega al destino. Los diseñadores de redes **IP** basan su proyecto original en el encaminamiento de paquetes de forma *best effort*, un enfoque que no distingue explícitamente las necesidades de flujos particulares. Las tentativas de solución de esta aparente dicotomía envuelven muchas tecnologías y filosofías conocidas simplemente como Calidad de Servicio (**QoS**), que, en una red **IP**, definen la habilidad de la red compensar características diferentes de tráfico, sin comprometer el *throughput* medio de la red. Muchos grupos del **IETF** (*Internet Engineering Task Force*) están trabajando con diversos enfoques para estandarización de este servicio. Estos enfoques pueden resumirse en 2 grandes categorías: Servicios Integrados (**IntServ**) cuya filosofía básica es la reserva de facilidades a flujos individuales y Servicios Diferenciados (**DiffServ**) cuya filosofía básica es la priorización de los flujos pero organizados en clases. Por otro lado, una técnica nueva, llamada de **MPLS** (*Multiprotocol Label Switching*), que tiene mecanismos propios para proveer **QoS** y utiliza conmutación de etiquetas para el encaminamiento de paquetes, fue definida en la **RFC** 3031 del **IETF**. Esta tesis trata justamente de conceptos envueltos en implementación de **QoS** con esta nueva técnica, incluyendo una evaluación practica del desarrollo de flujos **TCP** en redes **MPLS**, a través de mediciones de parámetros de *throughput* y retraso en diversas circunstancias de la red. Para esto fueran montadas dos topologías de red. Una para evaluación de efectos de reducción de congestión cuando son usados recursos de ingeniería de tráfico y otra para verificación y evaluación de una efectiva diferenciación de servicio, con la utilización de recursos de Calidad de Servicio y **CoS** (*Class of Service*).

ABSTRACT

Due to the expansion of the Internet phenomenon and consequently of the **IP** *phenomenon, a new breed of traffic has emerged of that form that challenges the fundamental assumptions about the* **IP** *design of networks. This traffic, real-time multimedia streams, not only demand higher end-to-end delivery guarantees tan other data. On the other side,* **IP's** *developers based its original design on best effort packet forwarding, an approach that does not distinguish explicitly between the needs of particular streams. The attempts at resolution of this apparent dichotomy consist of many technologies and philosophies known as Quality of Service (***QoS***). In an* **IP** *network,* **QoS** *defines the ability to compensate for traffic characteristics without compromising average throughput. Several Internet Engineering task Force Groups are working on standardization approaches for* **IP**-*based technologies. These approaches fall into 2 categories: Integrated Services (***IntServ***) using reservation, and Differentiated Services (***DiffServ***) using prioritization. One new technology,* **MPLS** *(Multiprotocol Label Switching) with mechanisms for providing* **QoS** *and uses "Labels" for packets forwarding: is defined in* **RFC** *3031 of the* **IETF**. *The aim of this work is the implementation of* **QoS** *with this new technology including a practical experimentation of performance of* **TCP** *flows on the* **MPLS** *network through measurements of throughput and latency. For doing this we include the installation of a practical testbed with 2 different topologies. One for performance evaluation of the effective service differentiation when using resources of* **MPLS CoS** *(Class of Service).*

CAPITULO I

PLANTEAMIENTO INICIAL DE LA INVESTIGACION

1.1 Identificación del problema

Calidad de Servicio significa disponer de mecanismos que permitan controlar los parámetros de retraso (*delay*), variaciones de retraso (*jitter*), pérdida de paquetes (*packet lost*) y ancho de banda (*bandwidth*), que puedan ser ofrecidos a flujos individuales de tráfico o a clases de tráfico diferentes.

Retraso representa el tiempo que un paquete demora en llegar desde su origen hasta su destino.

Jitter es la variación del retraso del paquete. Dependiendo de la aplicación, no es muy importante el valor en si, sino la magnitud de variación.

Pérdida de paquetes, como su nombre indica representa el número de paquetes que por cualquier motivo no llegaron a su destino.

Ancho de banda es la banda necesaria por tipo de *codec* utilizado.

Todos estos parámetros son conocidos como medidas de **QoS** y dependiendo de las aplicaciones, estos toleran valores y parámetros unos más que otros. Específicamente, este trabajo focaliza la implementación de "Calidad de Servicio", agrupando las aplicaciones en clases (**Diffserv**), usando la estructura **MPLS**.

1.2 Formulación del problema

El gran problema es como ofrecer una buena Calidad de Servicio en la *Internet*.

El desarrollo cada vez más creciente de aplicaciones avanzadas, hacen lo mencionado anteriormente, en algo cada vez más crítico, y convierte en imperativa la necesidad de implementar mecanismos en la *Internet*, que mejoren el "desempeño" de la red, utilizando inclusive, nuevas tecnologías que le posibiliten soportar estas nuevas aplicaciones y ofrecer tales posibilidades controlando su comportamiento de acuerdo con las necesidades de las

aplicaciones. Los mayores usuarios de estos mecanismos serian, tanto las empresas que realizan negocios en la *Web* (para una mejora de su contenido/servicios), lo mismo que los proveedores de servicios de *Internet* (**ISPs**), que necesitan ofertar servicios con valor agregado para incrementar sus ventas. En otras palabras, se torna imperativa la existencia de otros servicios de mejor calidad que el actual *best effort.*

Los mecanismos mencionados que miden las condiciones de la red continuamente, y toman medidas de ajuste de los valores predeterminados, procurando mantener la calidad dentro de ciertos valores, es lo que se llama Calidad de Servicio o en inglés, **QoS**.

El problema general es: ¿Cómo se implementa una estructura **DiffService**? y el problema específico es: ¿Cómo se implementa una estructura **DiffService** con **MPLS** (*Multiprotocol Label Switching*)?

1.3 Objetivos de la investigación

Los objetivos de la presente investigación son:

1.3.1 Generales

a) Adquisición de conocimientos detallados de las tecnologías **QoS**.
b) Contribuir a mejorar la calidad de servicio en el uso de la *Internet.*
c) Promover la cultura del uso de las tecnologías de comunicaciones con calidad de servicio.

1.3.2 Específicos

a) Implementación de **QoS** en las comunicaciones vía *Internet.*
b) Explicar los conceptos de implementación de las estructuras **Diffserv** a través de mecanismos de la técnica **MPLS**.
c) Estudio detallado de los diversos mecanismos de colas en *switches/routers.*
d) Experiencias simuladas de los principales recursos del **MPLS**.

1.4 Justificación

En los últimos 15 años, se han realizado esfuerzos considerables para el desarrollo de arquitecturas de **QoS** (*Quality of Service*) para redes **IP,** procurando proporcionar niveles distintos de calidad a diferentes tipos de tráfico. A pesar de

esto, en la *Internet* actual no se ha difundido mucho la aplicación de ninguna de estas soluciones de **QoS.** Se continua ofreciendo el tipo de servicio inicial denominado "*best effort*" a todo tipo de tráfico.

El acceso a *Internet* posibilita el uso de múltiples aplicaciones desde un terminal remoto basado en texto, pasando por la navegación *Web* o Correo Electrónico, hasta complejas aplicaciones interactivas multimedia. La calidad del servicio de acceso a Internet depende, por lo tanto, de la calidad percibida en cada una de las diferentes aplicaciones en ejecución, y para poder trabajar en este ambiente en optimas condiciones, faltan concluir todavía investigaciones sobre el nivel de satisfacción de los usuarios en cada una de las aplicaciones individuales que se utilizan en la *Internet*, y sobre todo, como las distintas percepciones individuales inciden en la calidad percibida globalmente en el acceso al servicio.

La realización de este trabajo de investigación se justifica por su:

1.4.1 Naturaleza:

La *Internet* es una herramienta de gran beneficio para la humanidad, dado que permite la comunicación y la transmisión de información, entre usuarios que se encuentran separados por grandes distancias.

No existen investigaciones validas, que permitan determinar la Calidad de Servicio (**QoS**) por parte de los usuarios de la red. El tráfico es procesado en forma secuencial sin poder garantizar en forma efectiva, el tiempo de entrega y la calidad de la información que llega.

El incremento de los usuarios, lo mismo que el aumento de las aplicaciones avanzadas que se están ejecutando, están provocando una mayor degradación del servicio.

1.4.2 Magnitud

El desconocimiento de los usuarios, sobre la forma de determinar la Calidad de Servicio del acceso y utilización de la *Internet*, está generalizado.

Son muy pocos los usuarios que conocen los términos: *delay, jitter, packet lost* y *bandwidth*, que pueden ser ofrecidos para el manejo del flujo de paquetes.

1.4.3 Trascendencia

El acceso a la *Internet* posibilita a los usuarios el empleo de una serie de aplicaciones y el acceso a una gran cantidad de información. Su utilización es universal, y en la medida de que sus servicios sean cada vez de mejor calidad, muchos mas usuarios la utilizaran para acceder a la información que necesitan, o para establecer una comunicación valedera con otros usuarios.

1.4.4 Vulnerabilidad

Este tema de investigación es vulnerable, ya que es posible realizar pruebas de evaluación de *performance* de las características del **MPLS** empleando el simulador *OPNET IT GURU*. Una vez que se comprueben los valores y se obtengan resultados válidos, podrán ser utilizados en forma general.

1.5 Limitaciones y facilidades

Estas se refieren a los factores que pueden ocurrir para frenar u obstaculizar el desarrollo o aplicación del proyecto, entre estas se pueden mencionar:

Para la aplicación del proyecto se tomaran en cuenta los siguientes argumentos:

a. Los conocimientos sobre redes de computadoras y las maneras de comunicarlas, nos llevan a preguntarnos si es posible mejorar la manera en que los usuarios de la *Internet* puedan acceder a información en forma segura y rápida, estableciéndose prioridades en la ejecución de los pedidos. La respuesta a esta pregunta es: SI.
b. Se utilizara para este trabajo el conjunto de protocolos **TCP/IP** que permiten que se comuniquen las computadoras y/o cualquier otro terminal. Se utilizara el modelo de redes **TCP/IP** que comprende cinco capas.
c. Se efectuaran pruebas de evaluación de *performance* de algunas de las características del **MPLS** utilizándose el simulador de red *OPNET IT GURU*.

1.6. Hipótesis de partida

En el tipo se servicio conocido como "*best effort*" para *Internet*, el tráfico de datos se procesa de acuerdo a como llegan las solicitudes de atención y sin ofrecer una garantía que la atención va a ser posible, y menos aun, el tiempo de demora. El presente trabajo de tesis, parte de la necesidad de mejorar la calidad de servicio

para los usuarios de la *Internet* en función de la aplicación que esté siendo usada. Se presume que la implementación de recursos de **QoS** a las aplicaciones agrupadas en clases, tendrá sus parámetros que alterarán la calidad de servicio, más o menos rígidas, de acuerdo con sus propias características.

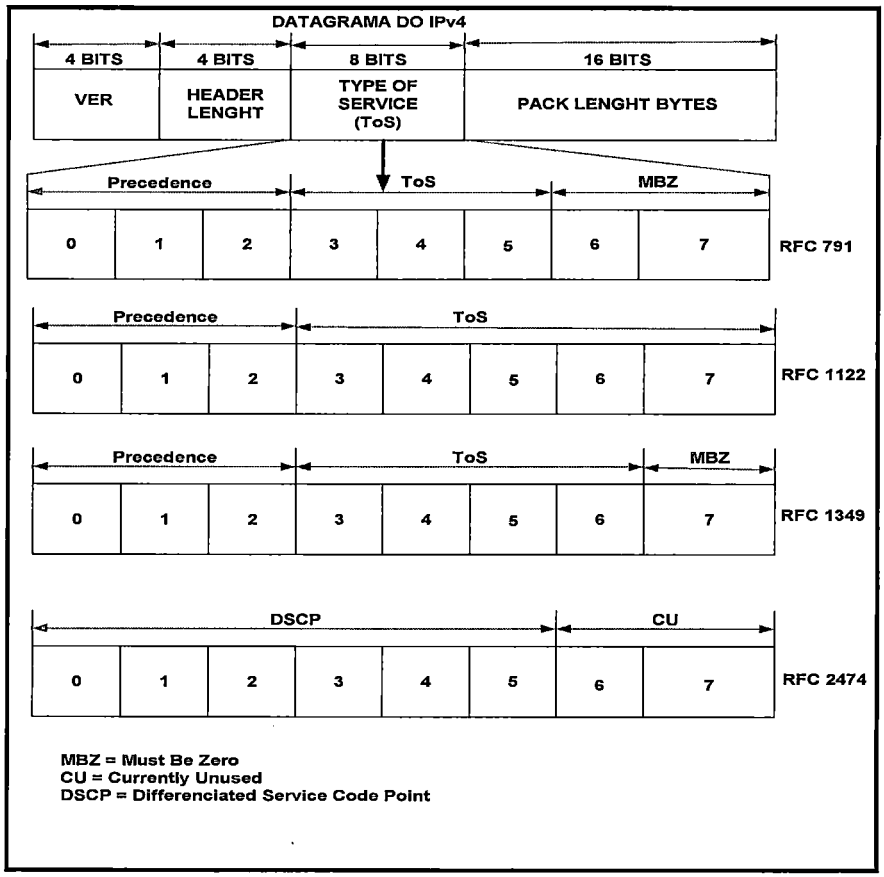
CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes del estudio

La preocupación de proporcionar más de una clase de servicio "*best effort*" no es reciente; al contrario, ha sido parte de la estructura **IP** (*Internet Protocol*) por más de 25 años. Podemos considerar como el primer gran paso en esta dirección la fecha de septiembre de 1981, cuando la norma **RFC** 791 (*Request for Comments*) estandarizo el protocolo **IP** reservando el segundo *byte* del cabezal **IP** para el campo Tipo de Servicio (**ToS**). Los *bits* de ese *byte* fueron definidos conforme indica la Fig. 2.1.

Fig.2.1: Estructura del ToS/DSCP.



ToS en **RFC** 791.

- *Bits* 0 – 2: *Precedence*
- *Bit* 3: 0 = Normal *Delay* 1 = *Low Delay*
- *Bit* 4: 0 = Normal *Throughput* 1 = *High Throughput*
- *Bit* 5: 0 = Normal *Reliability* 1 = *High Reliability*
- *Bits* 6 – 7: Reservados para uso futuro.

Desde aquella época, ya había una idea de cómo diferenciar retraso (*delay*), tasa de transmisión (*throughput*) y confiabilidad (*reliability*), y prever o seleccionar una prioridad relativa entre paquetes. Obsérvese también las modificaciones que este campo ha tenido desde las diversas **RFCs** hasta la actual **RFC** 2474, que trata de **DSCP** (*Diffserv Code Point*) de los servicios diferenciados, o **Diffserv** (*Differentiated Service*). La utilización de los *bits* de prioridad fueron solamente empleados a mediados de 1990 como soporte de la característica conocida como Descarte Selectivo de Paquetes (**SPD** - *Selective Packet Discard*). El **SPD** fija los *bits* de prioridad de forma tal que, si la red sufriera congestionamiento, el tráfico crítico (más importante), podría ser el último a ser descartado. Por otro lado, los *bits* **DTR** (*Delay*, *Throughput*, *Reliability*) nunca o casi nunca fueron utilizados.

Al final de los años 90, el **IETF** percibió que el **IntServ** no era adecuado para ser implementado en redes operativas y que sería, por tanto, mas adecuado hacer modificaciones para soportar servicio diferenciado a través de "clases de aplicaciones" para clientes específicos, o aplicaciones de una forma mas escalable, mediante la utilización de los *bits* de prioridad y creó el Grupo de Trabajo **DiffServ** (*Diffserv Working Group*). Entre otras cosas, el grupo cambio el nombre del octeto **ToS** del **IPv4** por *byte* **DSCP**, según se muestra en la Fig. 1.1. Esta nueva especificación del *byte* **DSCP** se aplica también al octeto de "Clase de Tráfico" del **IPv6** definido en la norma **RFC** 2475.

Una técnica relativamente nueva de enrutamiento de paquetes, que utiliza etiquetas y dispone de recursos para la implementación de **VPNs** (*Virtual Private Networks*), **TE** (*Traffic Engineering*) y **CoS** (*Class of Service*) fue definida por el **IETF** en la norma **RFC** 3031, y todo indica que sus recursos se relacionan

perfectamente a las necesidades de la estructura **DiffServ**. Esta técnica es conocida como **MPLS** (*Multiprotocol Label Switching*), considerando por un lado, la gran escalabilidad esperada por la estructura **DiffServ** para uso en la *Internet* actual, y por otro lado, las características del **MPLS**, dentro de las cuales está, la de proporcionar conmutación de paquetes de forma súper-rápida y recursos de Ingeniería de Tráfico. La combinación de los dos, o la implementación de **DiffServ** vía técnica de **MPLS**, lo tornan una estrategia muy atractiva para los instaladores de redes "*Backbone*".

2.2.- Definición de términos

ARCA: Analizador de Redes de Caminos Virtuales, *software* que analiza el problema de brindar garantías de Calidad de Servicio (**QoS**) así como realizar Ingeniería de Tráfico sobre redes de datos.

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

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

BGP: *Border Gateway Protocol*, protocolo de ruteo interdominio.

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

CLNS: **OSI *Connectionless Network Service***, *protocolo bajo el cual opera* **SNMP** v1.

CR-LDP: *Constraint Route* LDP, protocolo de distribución de etiquetas del tipo enrutamiento explícito que ofrece características de Ingeniería de Tráfico.

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

DDP: ***AppleTalk Datagram-Delivery Protocol***, protocolo bajo el cual opera **SNMP** v1.

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

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

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

IPX: ***Novell Internet Packet Exchange***, protocolo bajo el cual opera **SNMP** v1.

IS-IS: ***Intermediate System to Intermediate System***, protocolo de ruteo intradominio.

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

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

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

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

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

LSR: ***Lable Switch Router***, *router* encargado de dirigir el tráfico dentro de la red

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

MIRA: ***Minumum Interference Routing Algorithm***, algoritmo de ruteo de caminos que intenta minimizar la "interferencia" que provoca el establecimiento de un nuevo camino a potenciales nuevos caminos que son desconocidos.

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

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

MPLS: ***Multi Protocol Label Switching***, tecnología de ruteo y reenvío de paquetes en redes **IP** que se basa en la asignación e intercambio de etiquetas, que permiten el establecimiento de caminos a través de la red.

NET-TE: ***Networking Traffic Engineering***, nombre del *software* diseñado en este proyecto que hace alusión a la Ingeniería de Tráfico en redes; tema principal de éste trabajo.

NMS: ***Network Management System***, estación administradora o, lo que es similar, elemento de red que contiene un agente **SNMP** y pertenece a la red administrada.

QoS: ***Quality of Service***, distintos niveles de servicio que son ofrecidos al cliente en términos del ancho de banda o algún otro parámetro.

RSVP-TE: ***Reservation Protocol with Traffic Engineering***, protocolo de enrutamiento explícito. En éste caso en particular, es una extensión de la versión original **RSVP** que incorpora el respaldo para **MPLS**.

SDP: ***Shortest Distance Path***, ruteo basado en preservar los recursos de la red por medio de la selección de los caminos más cortos.

SLA: ***Service Level Agreement***, acuerdo sobre el nivel de servicio con el cliente donde se especifican parámetros como *performance*, confiabilidad y seguridad.

SMI: ***Structure of Management Information***, partes del **ASN**.1 que usan **SNMP**. Es lo que en realidad describe la estructura de datos del **SNMP**.

SNMP: ***Simple Network Management Protocol***, protocolo de la capa de aplicación que facilita el intercambio de información de gestión entre elementos de la red y es parte del *stack* de protocolos **TCP/IP**.

SWP: ***Shortest Widest Path***, ruteo que se basa en la búsqueda del camino con el ancho de banda más grande y, en caso de haber múltiples caminos, se queda con el que tiene la mínima cantidad de saltos.

TE: ***Traffic Engineering***, disciplina que procura la optimización de la *performance* de las redes operativas.

TED: ***Traffic Engineering Especialized Data Base***, base de datos contenida en cada *router*, la cual mantiene atributos de los enlaces de la red e información de la topología.

UDP: ***User Datagram Protocol***, protocolo de transporte que provee servicios de datagramas por encima de **IP**.

VPI/VCI: ***Virtual Circuit Identificator used in*** **ATM**, etiqueta o encabezado que puede utilizarse como etiqueta de **MPLS** en redes **ATM**.

WSP: ***Widest Shortest Path***, ruteo que se basa en la búsqueda de caminos con el mínimo número de saltos y, si encuentra múltiples caminos, se queda con el que tiene ancho de banda mayor.

CAPITULO III

METODOLOGÍA

3.1. Relación entre las variables de la investigación

- Variables independientes.

  IPDT = retraso

  IPDV = Variación del retraso

  IPLR = Pérdida de paquetes

- Variable dependiente

  QoS= Calidad de Servicio.

3.2. Tipo de investigación.

La investigación es del tipo descriptiva-experimental.

3.3. Diseño de la Investigación.

El proyecto prevé experiencias de simulación con las principales aplicaciones como FTP, voz sobre IP y videoconferencia.

3.3.1. Calidad de servicio en la Internet.

Durante el año del 1997, en varios *workshops* sobre *Internet2*, las cuestiones relacionadas con **QoS** se han debatido ampliamente de distintos ángulos y puntos de vista. La primera de ellas en *Ann Arbor* (**E.E.U.U.** [Estados Unidos de América]), en julio del 1997, se centró en requisitos de la red de aplicaciones estratégicas avanzadas y la necesidad de mecanismos que implementen **QoS**. El segundo *workshop*, en *Davis* (**EE.UU.**), en septiembre del 1997, abordó redes de campus y las relaciones con la administración y la ingeniería de proyectos de redes de campus. Un tercer *workshop* en San José (**EE.UU.**), en noviembre del 1997, fue destinado a diseñadores de *gigaPoPs* en análisis de actividades administrativas y de ingeniería, relacionadas con la extensión de **QoS** a través de *Internet2/gigaPoP*. El último *workshop*, en octubre del 1997, tuvo dos objetivos principales:

- En segundo lugar, discutir la parte administrativa y política de ramificaciones para aplicar **QoS** entre dominios.

Cabe mencionar también el intento de describir los requisitos de un centro de servicio **QoS**, en San José, en agosto del 1997, presentado por el "grupo de trabajo *ad hoc* de *Internet2*", con el objetivo de clasificar requisitos de **QoS** para *Internet2* y *multicast*, una iniciativa que dio lugar a presentación de un *Internet Draft* (**QoS** *requirements for Internet2*). Este documento trajo importantes contribuciones e información a los miembros de grupos de trabajo de *Internet2*, que se resume en este Capítulo.

a) Proyecto de un modelo de QoS

En nivel macroscópico, se puede considerar que el proyecto de un servicio **QoS** tiene tres componentes:

- Ámbito de Aplicación.
- Modelo de Control.
- Garantía de Transmisión.

- **Ámbito de Aplicación**

El ámbito de aplicación define límites (*boundaries*) de servicio **QoS**. Por ejemplo, en un alcance de fin-a-fin se puede acceder a las aplicaciones en sistemas terminales (*end systems*). Un ejemplo de alcance de fin-a-fin es una **RSVP** entre *hosts* para decidir un nivel de **QoS** pre-especificado. Además, otro ámbito de aplicación de servicio puede ser el intermediario, que no necesariamente permite acceso a sistemas finales.

- **Modelo de Control**

El modelo de control describe la granularidad y duración local de control de solicitudes de **QoS**, que se puede hacer desde puntos terminales o intermedios, y cuyos efectos pueden variar en duración y granularidad. La granularidad se puede extender de un simple flujo entre *hosts*, hasta un punto de agregación mientras que la duración puede extenderse desde un mes de *life time* de un simple flujo, hasta varios meses.

- **Garantía de Transmisión**

La garantía de transmisión se caracteriza por una granularidad, un conjunto de parámetros de transmisión, y las correspondientes garantías sobre lo que la red ofrecerá a cada uno, como en el modelo de control. La granularidad de una garantía de transmisión puede variar de un flujo único entre *hosts* hasta un local de nivel de agregación. Los parámetros de transmisión establecen las medidas definibles y configurables del modelo **QoS**. Entre los parámetros del modelo se puede citar: tasa de pérdida de paquetes, ancho de banda, retraso medio y variación, fiabilidad y **MTU** (*Maximum Transmission Unit*).

Las garantías de transmisión se caracterizan por dos factores:

Marco de Referencia (*Frame of Reference*).

Rigidez (*Rigidity*).

- **Marco de Referencia:** El marco de referencia específica dónde la garantía es relativa a otros flujos, o si será hecha en su totalidad, independientemente de otro tráfico.

- **Rigidez:** La rigidez determina como será la garantía, de modo suave (*soft*) o como probabilidad, por ejemplo, el máximo de 5% de sus paquetes serán retrasados por más de 30 ms. Con límites "rígidos" (*hard*), de modo que, si se produce una falla en la red, sus paquetes "nunca" serán retrasados por más de 30 ms.

b) Requisitos de QoS en *Internet2*

Cualquier infraestructura de la red para satisfacer las metas de **QoS**, entre otras, debe atender a las siguientes especificaciones:

- Permitir aplicaciones avanzadas.
- Admitir múltiples implementaciones interconectadas de equipos de enrutamiento de paquetes y nubes de redes.
- Tener buena escalabilidad.
- Ser administrable.
- Proporcionar un servicio mensurable.

- Trabajar con sistemas operativos de *end hosts* y *middleware*.

Obviamente muchos de estos objetivos son definitivos y de largo plazo, sin embargo debe empezar una implementación parcial y de corto plazo, a partir de una fecha, teniendo en cuenta estos objetivos finales.
Cualquier planteamiento debe ser aumentado en alcance y funcionalidad, sin embargo debe ser diseñado inicialmente para converger para la producción de una infraestructura con los objetivos anteriores.

- **Aplicaciones Avanzadas**

Este es un punto esencial, la razón de ser de *Internet*2. Actualmente no tiene sentido la funcionalidad de una red para un conjunto específico de aplicaciones. En cambio, las tecnologías de red de mayor éxito son aquellas que ofrecen servicio de bajo nivel (*low level*), como encaminamiento de paquetes, enrutamiento, transporte, etc., que trabajan para cualquier aplicación.

Hace veinte años, la aplicación más importante fue **RJE** (*Remote Job Entry*). Hace diez años la mayoría de aplicaciones utilizadas eran (y siguen siendo) **E-MAIL** (*Electronic Mail*), **TELNET** (*Teletype Network*) y **FTP** (*File Transfer Protocol*). Hoy, sin duda alguna, la aplicación más utilizada es la *web*. ¿Cuáles serán las aplicaciones en cinco o diez años?

Considerando este marco, la adopción de un plan de **QoS** para *Internet*2 debe tener en cuenta que este debe ser lo más general y duradero como las tecnologías principales de *Internet* como **IP**, **UDP** (*User Datagram Protocol*), **TCP**, **BGP** (*Border Gateway Protocol*), etc.

Considerando que no es conveniente diseñar una red en torno de necesidades específicas de aplicaciones actuales, es conveniente que los ingenieros de red conversen con los diseñadores de aplicaciones avanzadas para garantizar que las necesidades de estas nuevas aplicaciones sean consideradas explícitamente, en términos que correspondan a modelos viables de **QoS**.

- **Descripción de los requisitos de las aplicaciones:** Muchos diseñadores de aplicaciones, cuando se les pregunta sobre lo que necesitan de la red, por lo general contestan algo como lo siguiente: "bastante ancho de banda - la medida de lo posible - y con baja latencia, *jitter* y pérdida". Esta respuesta no es muy fuera de contexto si se considera el modelo "*best effort*", porque, con el tiempo, los desarrolladores de aplicaciones se han convertido en "expertos" de aplicaciones adaptativas, que simplemente funcionan mejor cuando la red es mejor. Sin embargo, no todas las aplicaciones avanzadas tienen o aceptan requisitos de manera vaga. Las necesidades de la mayoría de aplicaciones se basan en factores humanos o controles de tiempo real, por lo general una difícil adaptación. Generalmente es necesario un ancho de banda de unos pocos *megabits* por segundo y límites de latencia (*latency*) máxima entre 30 ms y 200 ms.
- **Punto de vista del usuario:** Del punto de vista del usuario, cuando este pide una clase de servicio, la calidad debe mantenerse de principio al fin para todos los componentes de la aplicación, desde texto, audio y video hasta cualquier otro. Por ejemplo, si la aplicación envuelve audio en tiempo real, la calidad de audio deberá ser lo mínimo equivalente a una transferencia a través de una línea telefónica convencional, y si la aplicación incluye video de *full-frame* y *full-motion*, ella deberá ser similar en calidad **VHS** (*Video Home System*), o **NTSC** (*National Television System Committee*), o mejor. En general, los usuarios saben que, en la implementación de un modelo de **QoS** no crea nada, el mismo ancho de banda, pero ellos esperan que, cuando estén bien establecidas las características de niveles de **QoS** que se ofrecerán, el comportamiento de la aplicación deberá ser claramente definido, por ejemplo, la clase C será de mejor calidad que la clase B y esta mejor que la clase A, o viceversa. También los usuarios esperan que se mantenga un servicio de calidad equivalente al "*best effort*". Las versiones actuales de **FTP**, **TELNET**, etc., deben continuar siendo utilizadas y tener acceso a servicios de red de calidad equivalente a las que existen hoy. Los usuarios también son concientes de la siguiente premisa: "pagar más para tener más" y de acuerdo con ella este pago se puede hacer en una sola negociación, que se define, por ejemplo, que el correo electrónico tendrá baja prioridad y los servicios multimedia alta prioridad, o como pago de tarifas específicas de acuerdo con la calidad de servicio solicitado. Esto, conduce básicamente a dos precios: una tarifa

flat, en la cual son predefinidas las reglas generales de juego, y otra, tasada por la conexión cada vez que una clase de servicio es solicitada. Los usuarios esperan tener acceso a los servicios de **QoS**, de dos formas básicas:

- Dinámica, cuando se desea utilizar una aplicación con **QoS** habilitado en este instante.
- Programada, cuando la planificación es hecha para un determinado día y horario. Por ejemplo, cuando se asiste un curso a distancia.

Los usuarios también son conscientes de que pueden recibir "señales de ocupado" al tratar de conectar a los servicios con mejor calidad de "*best effort*". Esperan que para las situaciones de degradación de los principales circuitos, puedan ser previstos circuitos "*fallback*", aunque de calidad inferior al principal, hasta que la situación se ha normalizado para restaurar la disponibilidad de canales de calidad requerida. Otro "*fallback*" podría ser utilizado para volver al circuito principal. En cuanto a eso, ellos no consideran prioridad la privacidad de datos (seguridad de aplicación); para ellos, ella no es vista como una facilidad de servicio **QoS** a ser ofrecida por la red. Sin embargo, esperan tener un mecanismo similar o "autenticación en la red" para la conexión al servicio de "mejor calidad que el actual *best effort*". Es importante que las reservas de recursos tengan efecto fin-a-fin, no hay requisito de que las reservas de recursos sean de esta forma. Es importante que la aplicación funcione de acuerdo a lo esperado. Por último, para los usuarios, las configuraciones cliente-servidor son también factores que afectan el rendimiento de la aplicación (ejemplo, congestión, *jitter*, etc.) y, por tanto, los administradores de aplicaciones tienen la responsabilidad de supervisar el sistema de sus clientes y de desactivar, si necesario, las aplicaciones que puedan interferir en la aplicación de **QoS** en curso. Al mismo tiempo, la expectativa es que los sistemas operativos de los servidores sean más sofisticados en relación a tramitación de las prioridades de los sistemas del cliente, y que el "*gap*" se reduzca aún más.

- **Punto de vista de los diseñadores de aplicaciones:** Los diseñadores de aplicaciones desearían, sobre todo, una manera padrón de "*feedback*" de la red, que podría incluir, entre otras cosas:

- Condiciones que permitan a la red contestar a una solicitud, de tal manera que la aplicación pueda negociar un conjunto más limitado de recursos, si necesario.
- Informaciones precisas de ocasiones en que la red no pueda mantener la atención a la solicitación que requieren, por lo tanto, una negociación o la adaptación de la aplicación.
- Informaciones sobre las ocasiones en que una solicitud está siendo atendida dentro de lo esperado.
- Posibilidad de acceso a las funciones de **QoS** a través de un conjunto definido de abstracciones (**API** [*Application Programming Interface*] y librerías), que permitan cambiar los mecanismos básicos y, en su caso, la filosofía general, sin necesidad del reemplazo total de la aplicación.

Deben agotarse los esfuerzos para medir los requisitos de sus aplicaciones. Incluso si las aplicaciones son adaptativas, es importante cuantificar los diferentes niveles de calidad. En el caso de aplicaciones adaptativas dentro de ellas deberán ser establecidos parámetros como bandas de velocidades, latencia, pérdida de paquetes, etc., de manera que funcionen de acuerdo con lo esperado.

- **Medida de Éxito:** A continuación, se presenta un resumen ilustrativo de la filosofía de evaluación de **QoS** en la *Internet2*, por parte de los usuarios y diseñadores de aplicaciones:

- Los usuarios podrán utilizar las aplicaciones actuales, al menos, con la calidad disponible en la actualidad (*Best Effort*).
- Los usuarios podrán utilizar las nuevas aplicaciones avanzadas, recibiendo garantías de ancho de banda y/o retraso, por la duración de toda la aplicación.
- Los usuarios podrán invocar aplicaciones con **QoS** habilitada, con un mínimo de "señales de ocupado".

En general, los usuarios de aplicaciones necesitan creer en la integridad de autenticación, autorización y en los mecanismos de contabilidad y en las reservas de recursos de **QoS**, adaptándose, de acuerdo a sus demandas, de **QoS**. Los diseñadores de aplicaciones deberán acezar las facilidades de **QoS** a través de

algunas abstracciones estándar (no primitivas en la capa de red) y tendrán acceso a herramientas que les permitan cuantificar sus necesidades en las aplicaciones.

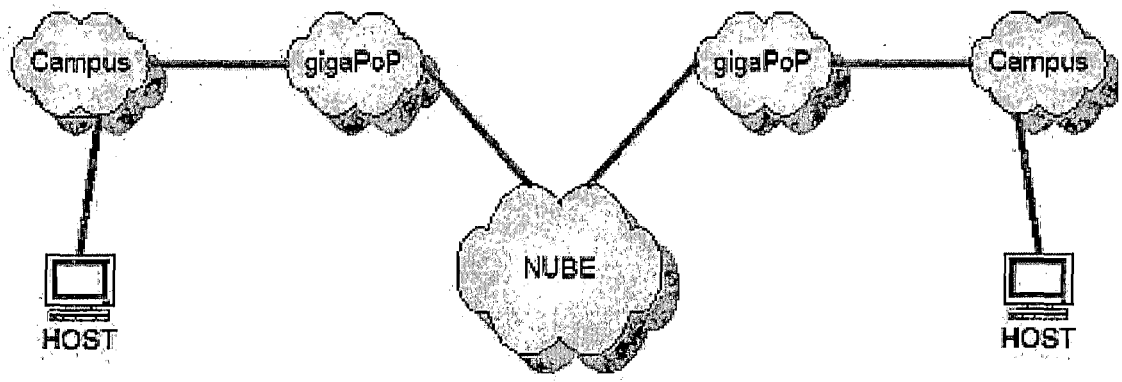
- **Interconectividad**

Cualquier enfoque para **QoS** en la *Internet2* debe admitir múltiples implementaciones interoperables de servicios y de elementos individuales de la red.

La Fig. 3.1 muestra una situación típica en el entorno de *Internet2*. Cada elemento de la red, desde el Campus, *gigaPoP* y "nube", cada uno por lo general bajo un control administrativo diferente, implica la necesidad de estandarizar el concepto de **QoS** a través de cada elemento, de manera que la composición de las "nubes" pueda proporcionar un servicio fin-a-fin, con pleno efecto en la red.

Interconectividad puede considerarse como un aspecto manejable de escalabilidad. Señalización y configuración de los flujos deberán ser tratados de una manera estandarizada y bien comprendida dentro de los límites de nube a nube, mientras que se les debe permitir a cada nube la implementación de **QoS** internamente, potencialmente de muchas maneras diferentes. Estas implementaciones internas pueden variar, dependiendo de la tecnología básica de la nube, de políticas internas y de decisiones necesarias.

Fig. 3.1: Interconectividad



- **Interoperabilidad de Equipos**

Cualquier enfoque de **QoS** adaptado para *Internet2* deberá ser implementado por lo menos por uno de los fabricantes de equipos. Es deseable, sin embargo, que sean muchos. En una red heterogénea del tamaño de *Internet2*, la demanda de interoperabilidad de los equipos de diferentes fabricantes es el único enfoque razonablemente admitido. Para garantizar el éxito de **QoS** en la *Internet2*, es muy importante adoptar una estrategia que sea alineada con la dirección de los organismos de "*standards*" de *Internet* como el **IETF**. Pensando de esta manera, es deseable que la implementación empiece antes que sean completados los trabajos "*standard*". En este sentido, las experiencias en curso en la *Internet2* serán muy valiosas y proporcionaran "*feedback*" a los *standards* en vigencia, incluso *feedback* de interoperabilidad y no interoperabilidad de implementaciones alternativas.

- **Escalabilidad**

Sin duda el mayor desafío de ingeniería requerido por la condición de ofrecer **QoS** por flujo fin-a-fin, es la solución de situaciones en que los flujos de tráfico de los "bordes" (*edges*) de una red pueden fácilmente sobrecargar los *routers* centrales. Este es un problema muy grave, sobre todo en los núcleos de la red dónde cientos de flujos pueden pasar a través de cada *router*, y una estructura no escalable probablemente no soportara eso por mucho tiempo. La solución o mitigación de estas situaciones es un imperativo en la implementación de **QoS**.

- **Administrabilidad**

Deberá haber mecanismos para asignar y contabilizar los servicios de **QoS** ofrecidos. Esos mecanismos deben operar de manera eficiente, dando a los usuarios rápido acceso a facilidades de **QoS**, sin cargas excesivas de planificación y operaciones en la red. En general, se debe soportar un conjunto flexible de políticas y minimizar la acción de tentativas de "robo" de servicios de **QoS**. Como cualquier otro servicio, es necesario controlar el acceso, incluyendo personas no autorizadas. Esto nos lleva, entre otras cosas, a la necesidad de autenticación de la identidad del usuario o institución que solicita el recurso. También se debe tomar una decisión administrativa para permitir o no el acceso (*admission control*) y, además, para tener en cuenta debidamente su uso.

Un mecanismo seguro deberá confirmar la identidad del solicitante del servicio. Una vez conocida, deberá ser tomada una decisión de control de admisión. Para esto deberá existir una política y medios para determinar la disponibilidad de recursos. Desde que los flujos pasan a través de múltiples dominios administrativos, el control de admisión puede ser o no una decisión local, siendo, por tanto, necesario configurar a través de múltiples dominios administrativos.

- **Mensurabilidad**

Desde que instituciones y eventualmente usuarios, tengan que pagar por las facilidades de **QoS** que reciben, deberán existir también medios para medir y auditar el desempeño de la red. Esto no significa solamente la disponibilidad de herramientas para medir, sino una buena comprensión de los parámetros de rendimiento de la red.

Los proveedores de red (*network providers*) pueden, por tanto, necesitar herramientas de medida que ayuden los ingenieros de red en las facilidades de **QoS**, o que operen como mecanismos de apoyo de control de admisión basado en mediciones.

- **Requisitos del *Host***

A largo plazo, los *host* deberán ser capaces de iniciar las solicitudes de **QoS** a través de sus propios flujos. Sin embargo, en el corto plazo, podrían ser configurados estadísticamente por conector, por número de puerto, por protocolo, etc.

Los *hosts* también deben poder acceder adecuadamente por ellos mismos o por sus usuarios en la red, para una correcta autenticación, autorización y contabilidad. Además, para proporcionar una verdadera **QoS** fin-a-fin, los sistemas operativos de los *hosts* deberán soportar flujos de **QoS** habilitados (**QoS** *enable flows*). Es importante recordar que actualmente, esta clase de funcionalidad *real-time* no está disponible en sistemas operativos de la mayoría de *hosts* de *Internet2* y, por tanto, los paquetes pueden experimentar embotellamiento dentro del "*stack*" de la red, en el sistema de memoria, o incluso dentro del programador de procesos.

- **Implementación Incremental**

No sería bueno adoptar un enfoque de implementación de **QoS** que se solicite, con la condición de uso, originando el cambio de todos los elementos de la red. Un enfoque más apropiado sería una implementación parcial, inicialmente en el punto de mayor congestión, lo que traería beneficios inmediatos a un número importante de aplicaciones existentes y que requieren **QoS** inmediatamente.

Otro importante argumento para la implementación parcial es el inicio de las ganancias de experiencia en esta área, sin que haya necesidad de esperar que los sistemas complejos estén plenamente disponibles. Esto permitiría que algunas facilidades ya disponibles pudieran hacer una reevaluación de prioridades y los plazos para determinadas actividades en función de los resultados obtenidos.

3.3.2- Técnicas para implementación de QoS

Normalmente el nivel de calidad de servicio de una red se mide por el grado de satisfacción de sus usuarios, sin embargo, también depende en gran medida de las aplicaciones en uso y de la percepción de los usuarios. En general, **QoS** representa el conjunto de mecanismos necesarios para gestionar y controlar los siguientes parámetros:

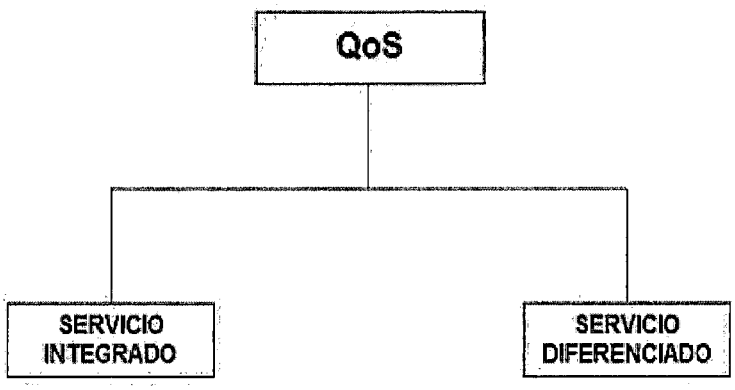
- Tráfico Transportado (*Throughput*).
- Retraso (*Delay*).
- Variación de Retraso (*Jitter*).
- Pérdida de Paquetes (*Packet Loss*).

La medición y la validación de estos parámetros no son tan sencillas, especialmente en el contexto de un fenómeno altamente subjetivo y variable como calidad de voz. **QoS** requiere la cooperación de todas las capas lógicas de la red **IP**, desde la capa de aplicación hasta la capa física y de todos los elementos de la red fin-a-fin. Obviamente no es fácil mantener el mismo rendimiento de **QoS** a través de todos los diferentes componentes de una red **IP** (fin-a-fin), entonces se presenta como una meta muy difícil.

El **IETF** define las siguientes estructuras para la prestación de **QoS** en la *Internet*:

- Servicios Integrados (**IntServ**), utilizando básicamente reserva de recursos - **RSVP RFC** 1633.
- Servicios Diferenciados (**DiffServ**) utilizando básicamente priorización - **RFC** 2475.

Fig. 3.2: Implementación de QoS por el IETF



a) Servicios Integrados - RFC 1633

Este servicio emula el concepto de asignación de recursos de conmutación de circuitos para proporcionar recursos de conformidad con las solicitudes de cada *host*.

El *host* inicializador envía los valores de tasa de transmisión necesaria, retraso o *jitter* requeridos. Los *hosts* (*routers*) intervinientes encaminan la solicitud al destino del *host* correspondiente.

Servicios integrados definen básicamente dos clases de servicio:

- Servicio de carga controlada - (*controlled - load service*): definido en la **RFC** 2211, este servicio asume "que un alto porcentaje" de los paquetes pasarán correctamente a través del *router* (sin desecho) y con un retraso de encolamiento de cerca a cero, pero no ofrece ninguna garantía cuantitativa de rendimiento. Además, depende en gran medida del estado de congestión de la red y, por tanto,

presenta un rendimiento ligeramente mejor que el servicio "*best effort*" y, por tanto, también llamado de "*best effort*".

- Servicio garantido (*Guaranteed Service*): definido en la **RFC** 2212, requiere límites precisos de retraso y *jitter* derivados del *link* de la red, incluso asignar el ancho de banda de acuerdo con la solicitud del *host* inicializador.

En general, la filosofía de esta técnica ofrece a los usuarios que tienen "*best effort*" condiciones de "carga baja" en la red, incluso si la "carga" aumente en la misma. Esto implica una condición *sine qua non*, que los *routers* sean capaces de reservar recursos para los diferentes flujos.

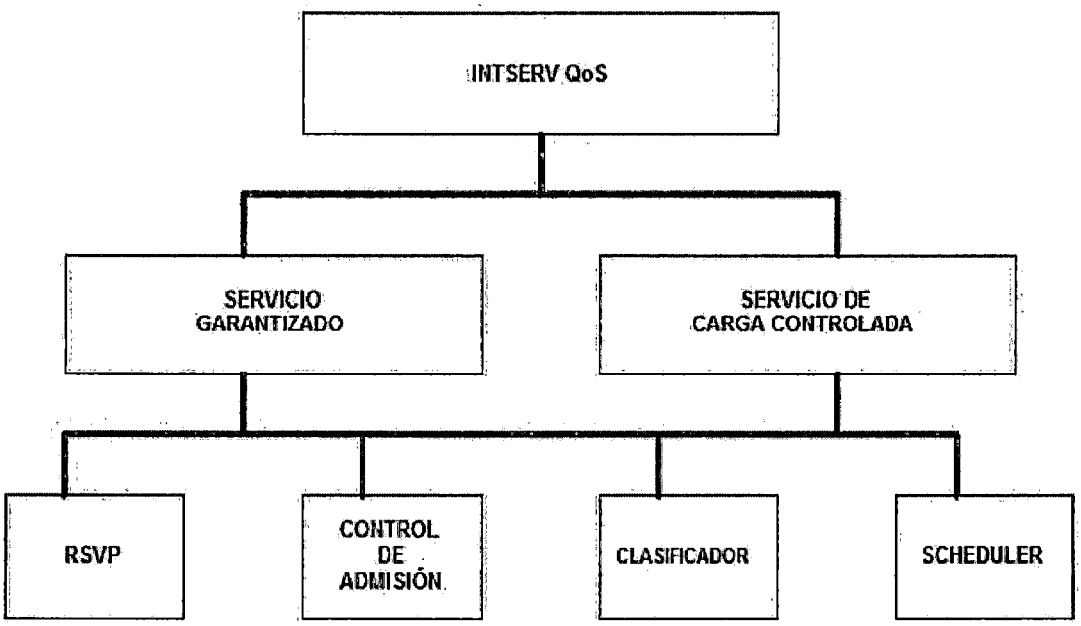
Cuando se habla de recursos en la *Internet*, suelen hablar de la banda en el *link* y "*buffers*" de los *routers*. El protocolo **RSVP**, definido en la **RFC** 2205, es el protocolo de señalización utilizado habitualmente en el establecimiento de reserva y control de recursos prescritos por servicios integrados. Los *hosts* y *routers* usan el **RSVP** para encaminar las peticiones de **QoS** a los *routers* al largo de los *links* y para "mantener" un estado que proporcione el servicio solicitado, por lo general ancho de banda y latencia fija. Para su implementación, es necesario que el *software* **RSVP** este instalado en los receptores, transmisores y *routers* de la red.

El servicio **IntServ** es implementado básicamente con 4 componentes:

- El protocolo de señalización (**RSVP**).
- La rutina de control de admisión.
- El clasificador.
- El "*scheduler*" de paquetes.

Las aplicaciones en las dos subclases (garantizada y controlada) de servicio, normalmente configuran sus conexiones y reservan recursos antes de transmitir sus datos. La rutina de control decide si una solicitud de recursos se puede garantizar. El clasificador hace una clasificación **MF** (*Multi-Field*) y pone el paquete en una cola especifica basada en el resultado de la clasificación. Por lo general, se usa **MF** (dentro de un filtro *firewall*) para determinar la clase de enrutamiento y prioridad de pérdida de paquetes. El *scheduler* de paquetes programa el paquete adecuadamente, de acuerdo con requisitos de **QoS**.

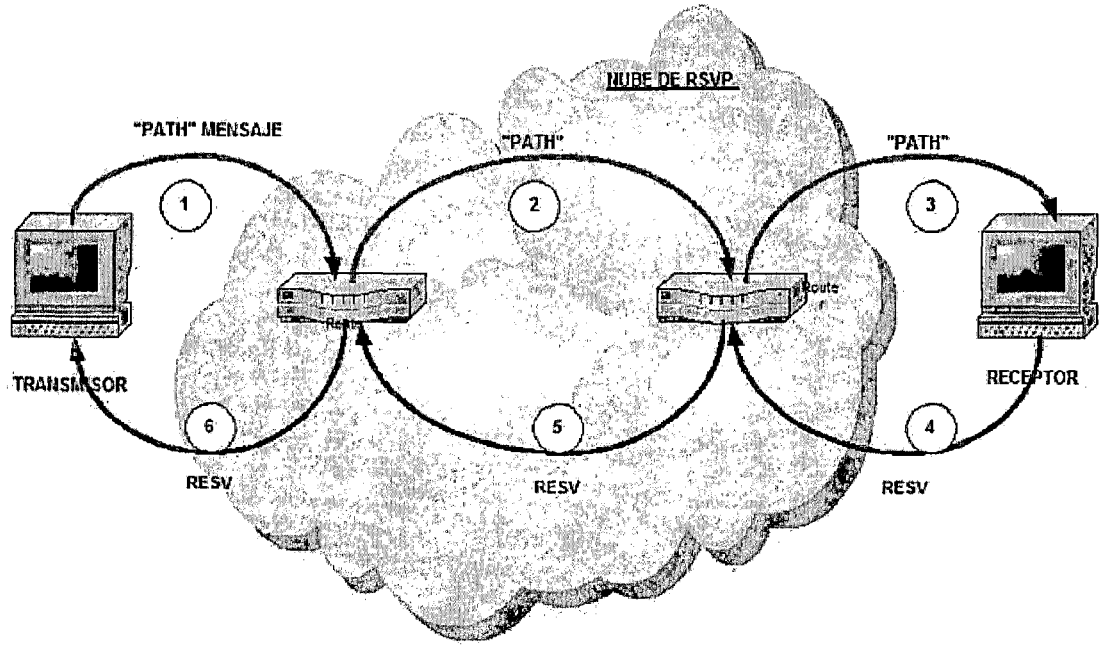
Fig. 3.3 Servicio Integrado



- **Funcionamiento básico del protocolo RSVP**

El procedimiento básico de funcionamiento es enseñado en la Fig. 3.4. El transmisor envía un mensaje "*path*" al receptor especificando sus características de tráfico. Cada *router* intermediario en el camino hacia adelante encamina el mensaje "*path*" para el siguiente *link* (determinado por el protocolo de enrutamiento). Cuando el receptor recibe el mensaje, responde con un mensaje **RSVP** y vuelve a los intermediarios de la ruta. Cada *router* intermediario puede aceptar o rechazar la solicitud de **RSVP**. Si la solicitud es rechazada, el *router* enviara un mensaje de error para el receptor y el proceso de señalización termina. Si la solicitud de **RSVP** es aceptada, el ancho de banda y espacio de *buffer* son asignados para el flujo y la información del estado de este flujo se almacenara en el *router*.

Fig. 3.4: Funcionamiento básico de RSVP



b) Servicios Diferenciados - RFC 2475

Este servicio, más conocido como **DiffServ**, fue presentado como resultado de la dificultad de implementación e instalación de servicio **IntServ** y de **RSVP**, especialmente en lo que se relaciona con escalabilidad. Entre las principales dificultades podemos mencionar:

. **IntServ** no es escalable, lo que significa aumentos significativos de estados por flujo y de procesamiento de paquetes en cada nodo a lo largo de la conexión fin-a-fin. En la ausencia de agregación de estados, el número de estados que necesita ser mantenido en cada nodo es proporcional al número de reservas simultáneas a través de un cierto nodo. Así, el número de flujos en un *link backbone* de alta velocidad podría potencialmente ir de docenas o cientos, a más de un millón.

- **IntServ** también requiere que las aplicaciones soporten el protocolo de señalización **RSVP**, que en el momento de su introducción, muy pocos sistemas operativos lo permiten.

Cuanto al protocolo **RSVP**, se puede decir que, si bien es cierto que la estructura **IntServ** en su conjunto puede ser considerada como un fracaso, no sucede lo

mismo con el **RSVP**, que, en ese momento, fuese desarrollando y cuya instalación se incremento, a punto que hoy es utilizado como protocolo de señalización de uso general para **MPLS** o de restablecimiento rápido de **LSPs** (*Label Switching Paths*). Funciona bien con **MPLS**, pues no presenta los problemas de escalabilidad de estructura **IntServ**.

La esencia del servicio **DiffServ** es clasificar el tráfico en varias clases, con un trato diferente para cada una de ellas, especialmente en situaciones de escasos recursos (congestión), y, a través de mecanismos capaces de proporcionar la escalabilidad necesaria. Como se mencionó anteriormente, el fracaso del modelo **IntServ** fue debido a explosión de señalización y al aumento de los estados por flujo que era necesario mantener en cada nodo de la ruta del paquete. En este servicio, se clasifica el comportamiento por *link* como un "Punto de Código de Servicio Diferenciado (**DSCP**)" y se usa los tres *bits* más a la izquierda del *byte* de **ToS** para indicar el tratamiento que el paquete debe de tener en su ruta. Elaboración de diferentes formas de clasificación, vigilancia, "*shaping*" y la programación, se pueden ofrecer para algunas clases de servicio en el nivel más bajo, alto *throughput* o baja pérdida, sin embargo las opciones son limitadas.

Este enfoque representa una forma de **QoS** suave (*soft*), de clasificación de los servicios a través del marcado de paquetes. Normalmente el **DSCP** de valor 46 (Cuadro 3.1) reduce los retrasos y el *jitter* y proporciona el mayor nivel de **QoS** agregado. Obviamente esta **EFC** (*Expedited Forwarding Class*) como se conoce esta clase, depende, entre otras cosas, de la implementación basada en *link* por *link* (*hop by hop*).

Una típica estructura **DiffServ** ofrece dos clases de servicio, más allá del básico "*best effort*". Ellos son: Servicio Garantizado (*Assured Service*) y Servicio *Express* (*Expedited Service*).

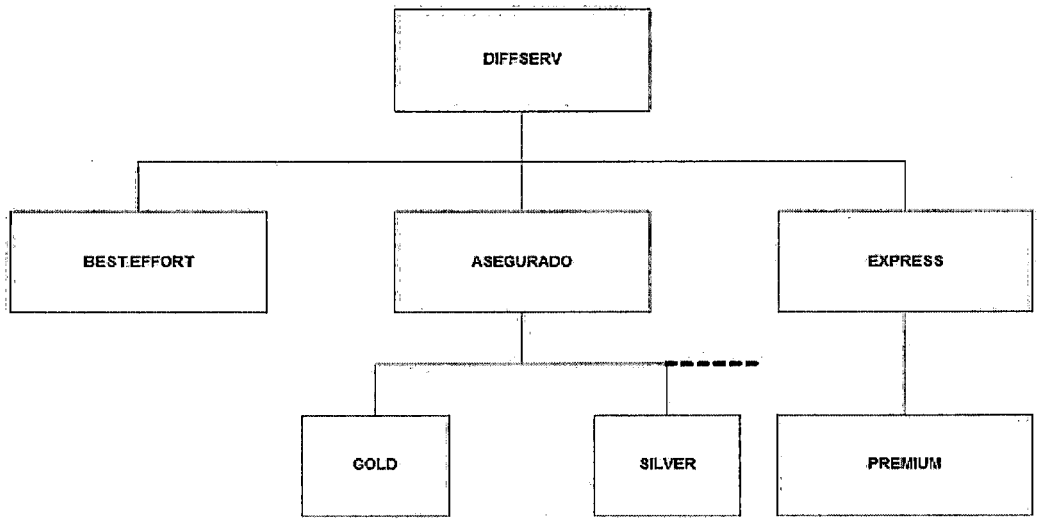
El servicio garantizado ofrece un servicio fiable y predecible de entrega de paquetes, dirigido principalmente a aplicaciones interactivas no *real-time*, como *web browsing*, por ejemplo. Los usuarios de este servicio por lo general tienen un contrato con los proveedores, conocido como **SLA** (*Service Level Agreement*),

que especifica, entre otras cosas, el tamaño de ancho de banda contratado y el retraso.

Cuanto al servicio *express* es un servicio fiable con baja demora y bajo *jitter*, similar al servicio obtenido a través de una línea privada. Está dirigido principalmente a aplicaciones en tiempo real. En este caso, también debe tener un **SLA** con el proveedor, especificando, entre otros la tasa media de *bits* y máxima deseada. El tráfico que supera este pico puede ser desechado o tolerado al costo de una tasa extra, decidido sobre la base de recursos de la red en aquel momento.

Para indicar la clase de tráfico, sólo los tres primeros *bits* de **DSCP** son utilizados (*bits* 0, 1 y 2 para prioridad) y, para indicar la preferencia o prioridad de rechazo, los dos siguientes (*bits* 3 y 4). Los otros tres últimos *bits* deben estar en cero (*bits* 5, 6 y 7).

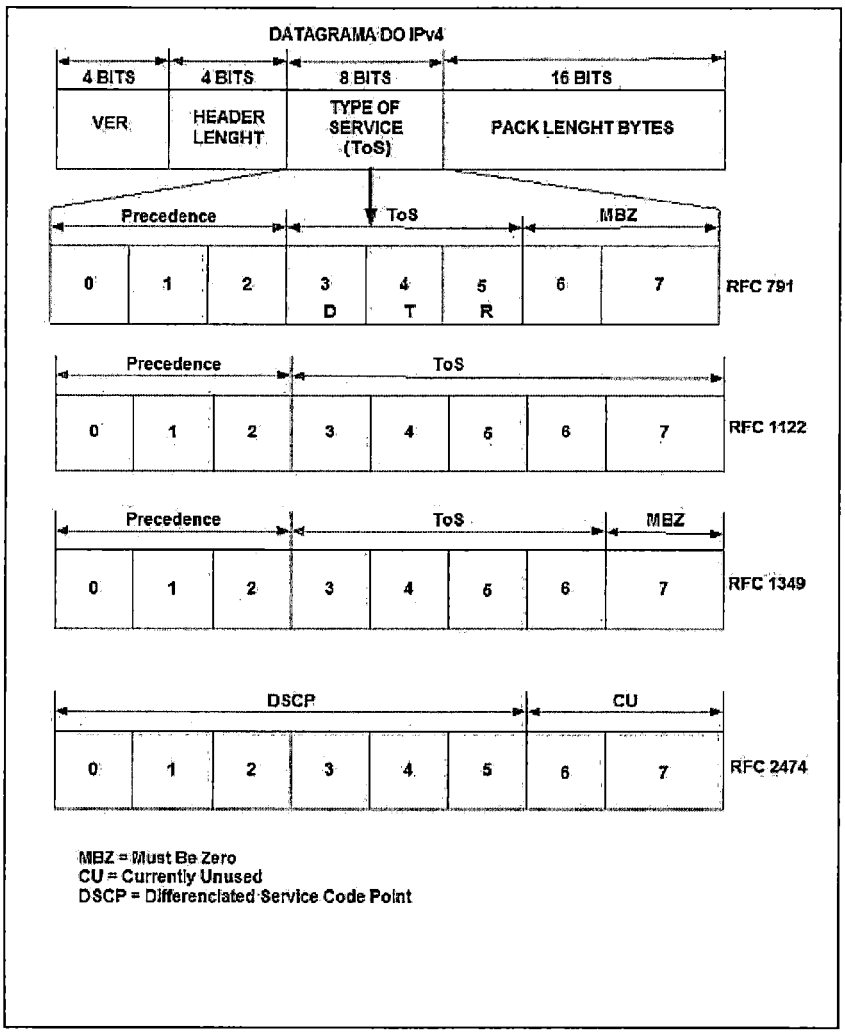
Fig. 3.5: Servicios Diferenciados



Cuadro 3.1: Clases DiffServ

Clase			Prioridad					PHB	Decimal	Prioridad
0	1	2	3	4	5	6	7			
0	0	0	0	0	0	0	0	*best effort*	0	
0	0	1	0	1	0	0	0	AF11	10	baja
0	0	1	1	0	0	0	0	AF12	12	mediana
0	0	1	1	1	0	0	0	AF13	14	alta
0	1	0	0	1	0	0	0	AF21	18	baja
0	1	0	1	0	0	0	0	AF22	20	mediana
0	1	0	1	1	0	0	0	AF23	22	alta
0	1	1	0	1	0	0	0	AF31	26	baja
0	1	1	1	0	0	0	0	AF32	28	mediana
0	1	1	1	1	0	0	0	AF33	30	alta
1	0	0	0	1	0	0	0	AF41	34	baja
1	0	0	1	0	0	0	0	AF42	36	mediana
1	0	0	1	1	0	0	0	AF43	38	alta
1	0	1	1	1	0	0	0	*Expedited*	46	

Fig. 3.6: Estructura de ToS/DSCP



ToS en la **RFC** 791.

- *Bits* 0 - 2: *Precedence*
- *Bit* 3: 0 = *Normal Delay* 1 = *Low Delay*
- *Bit* 4: 0 = *Normal Throughput* 1 = *High Throughput*
- *Bit* 5: 0 = *Normal Reliability* 1 = *High Reliability*
- *Bits* 6 - 7: Reservados para uso futuro

Cuadro 3.2: *Precedence* RFC 791

0	1	2	***Precedence***
1	1	1	*Network Control*
1	1	0	*Internetwork Control*
1	0	1	*Critic*/ECP
1	0	0	*Flash Override*
0	1	1	*Flash*
0	1	0	*Immediate*
0	0	1	*Priority*
0	0	0	*Rout Line*

ToS RFC 1122: cambió para incluir: *bits* 3 4 5 6 7

ToS RFC 1349: cambió para incluir: *bits* 3 4 5 6

***Bit* 7:** en cero

Cuadro 3.3: ToS RFC 1349

0	1	2	3	**Descripción**
1	0	0	0	*Minimize Delay*
0	1	0	0	*Maximize Throughput*
0	0	1	0	*Maximize* fiabilidad
0	0	0	1	*Minimize Monetary Cost*
0	0	0	0	*Normal Service*

- **Terminología**

Siguen algunos términos utilizados con frecuencia en MPLS:

Cuadro 3.4: Terminología

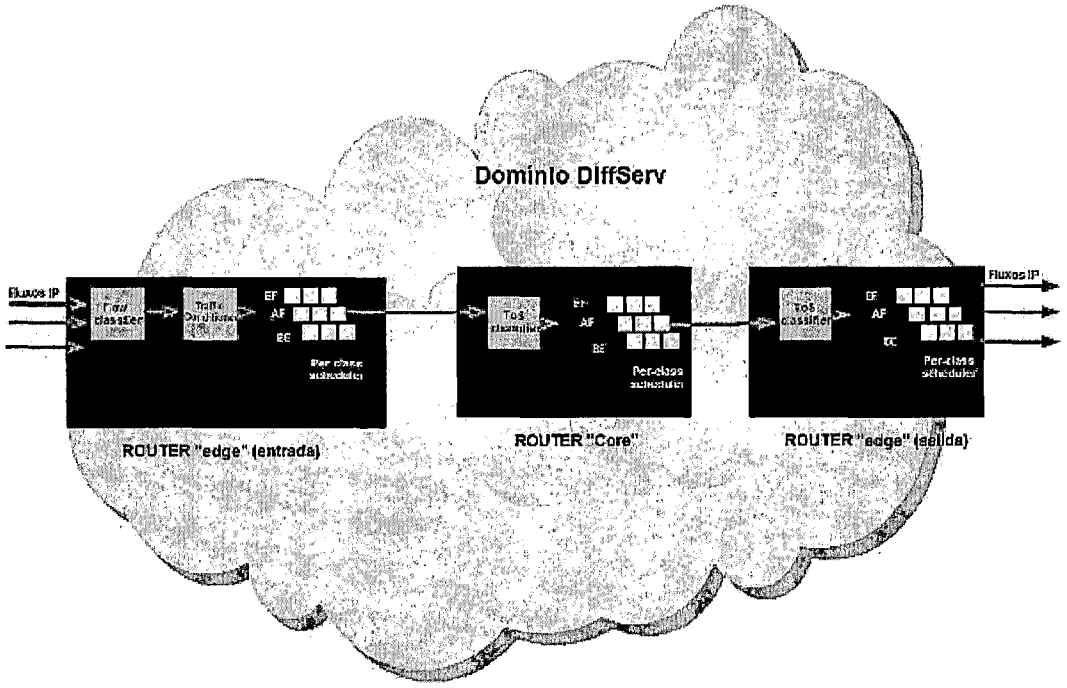
Flow	**Una secuencia de paquetes con la misma dirección IP de origen, puerto de origen, puerto de destino y protocolo ID**
Service Level Agreement (SLA)	Un contrato entre un cliente y un proveedor de servicio que especifica el servicio que el cliente deberá recibir. El cliente puede ser una organización u otro proveedor de dominio
Traffic Profile	Una descripción de las propiedades de un flujo de tráfico igual en velocidad y tamaño de *Burst*
Precedence Field	Los 3 *bits* más a izquierda en el octeto ToS de un encabezado IPv4. Vea que en DiffServ, eses 3 *bits* pueden o no utilizarse para denotar la prioridad de paquete IP
ToS *Field*	*Bits* 3 - 5 en el octeto ToS del encabezado IPv4
Differentiated Service field (DS *field*)	El octeto ToS de un encabezado IPv4 o el octeto clase de tráfico de un encabezado IPv6, renombrado en el servicio DiffServ. Es el campo dónde las clases de servicio son codificadas
Per-Hop-Behavior (PHB)	El tratamiento de encaminamiento externamente observable de una clase de paquetes en un nudo con servicio DiffServ
Mechanism	Un algoritmo u operación especifica

	(disciplina de encolamiento) que es implementada en un *router* para realizar un conjunto de PHBs
Admission Control	El proceso de decisión de aceptar una solicitación de recursos (ancho de banda y espacio de *buffers*)
Classification	El proceso de selección de paquetes basado en el contenido de los encabezados de los paquetes de acuerdo a las reglas definidas
Behavior Aggregate (BA) *Classification*	El proceso de selección de paquetes basado sólo en el contenido del campo DS
Multi-Field (MF) *Classification*	El proceso de clasificación de paquetes basado en el contenido de campos múltiples como dirección de origen, destino, *byte* ToS, protocolo ID, puerto de origen y puerto de destino
Marking	El proceso de colocación del campo DS de paquetes
Policing	El proceso de tratamiento de tráfico fuera de perfil (desecho de paquetes en exceso)
Shaping	El proceso de retrasar paquetes dentro de un flujo de tráfico para atender a alguno perfil definido de tráfico
Scheduling	El proceso de decidir cual paquete enviar primero en un sistema de colas múltiples
Queue Management	Control del tamaño de colas de paquetes por el desecho de paquetes cuando necesario o adecuado

- **Los componentes de una estructura de servicios diferenciados (DiffServ)**

La **RFC** 2475 no define "como" esta arquitectura deberá ser implementada. Indica sólo los componentes funcionales que deberá tener y que son los siguientes:

Fig. 3.7: DiffServ



1. *Routers* de Borde (*Edge Routers*), con funciones de clasificación de paquetes y acondicionamiento de tráfico.
2. *Routers* de Núcleo (*Core Routers*): con funciones de enrutamiento de los paquetes.
3. PHB (*Per Hop Behavior*): que define los diferentes rendimientos entre las clases. (No impone un mecanismo específico para alcanzar los objetivos. Cualquier técnica puede ser utilizada, a condición de que sea "externamente" observable y medida). Son ejemplos de PHB:

PHB1: 99% de los 2 Mbps para el 99% del tiempo.

PHB2: PHB1 + baja prioridad.

PHB3: PHB1 + alta prioridad.

El marcado de los paquetes se realiza en el campo DS del IPv4 o en el encabezado del paquete IPv6.

- **Modelos de PHB**

Actualmente se definen dos modelos de **PHB**:

1. Enrutamiento Expreso (*Expedited Forwarding*): definido en la **RFC** 2598 especifica que la velocidad de salida del *router* de una determinada clase de tráfico debe ser igual o superior a la velocidad configurada. Esto implica una garantía de suficiente ancho de banda, independientemente de la intensidad de tráfico de otras clases que lleguen en el *router*. Es muy común referirse a esta clase de tráfico como *premium*.
2. Enrutamiento Garantizado (*Assured Forwarding*): definido en la **RFC** 2597 divide el tráfico en 4 clases, cada una con un mínimo de *buffers* y ancho de banda. Cada una de estas clases se subdividen de acuerdo a 3 preferencias para la eliminación de paquetes, utilizadas en situaciones de congestión. En la literatura de **QoS** encontramos el **PHB** utilizado para indicar clases como: *gold*, *silver*, bronce, *platinum*, etc.

- **Escalabilidad del servicio DiffServ**

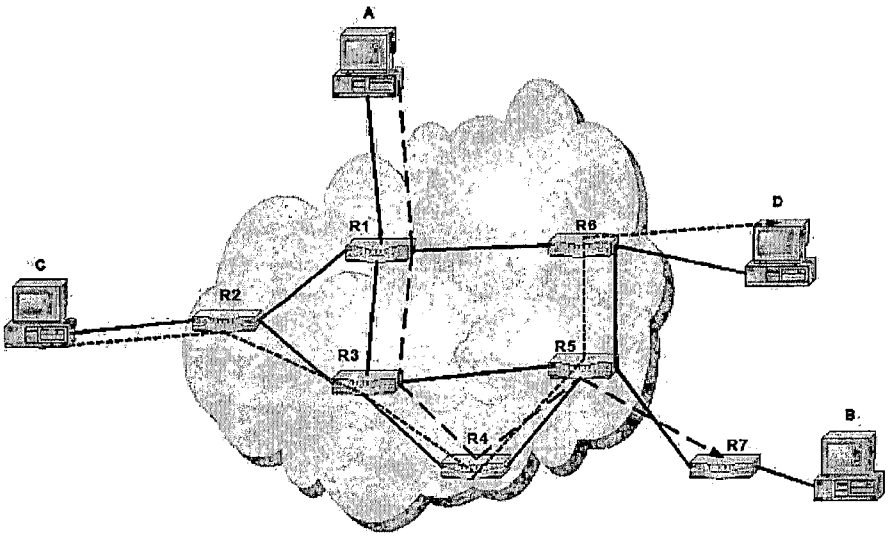
Con referencia a la Fig. 3.8, se puede observar:

1. Los paquetes de A para B serán marcados en R1.
2. Los paquetes de C para D serán marcados en R2.
3. Las marcas que reciben están de acuerdo con el tráfico al cual pertenecen. Diferentes clases de tráfico recibirán los distintos servicios en el núcleo de la red.
4. Un paquete se envía al siguiente *router* de acuerdo con el llamado "PHB".

Asociado con cada clase de paquete, el PHB define, entre otras cosas, como el ancho de banda y *buffers* son compartidos entre las diferentes clases de tráfico. Así, el enrutamiento es basado únicamente en las marcas en el paquete. En el caso de la Fig. 3.8, si los paquetes de A para B y los paquetes de C para D reciben las mismas marcas, R3 los tratara como un agregado sin tener en cuenta su origen, y por lo tanto no tiene que preocuparse con los flujos individuales

origen-destino, consiguiendo, por lo tanto, una buena escalabilidad, en concordancia con los objetivos propuestos.

Fig. 3.8: Escalabilidad



c) Un ejemplo de implementación de servicio "expreso y garantizado con DiffServ"

Se trata, con fines ilustrativos, únicamente las siguientes actividades en los *routers* de borde ("*edge*").

- Clasificación de los paquetes.
- Vigilancia, señalización y "*shaping*" de los paquetes.
- Gestión de filas y "*scheduling*" de los paquetes.

- **Clasificación de Paquetes**

La clasificación de los paquetes se basa principalmente en la *interface* de entrada. Esta *interface* puede ser física, tales como **PoS** (*Packet over Sonet*) lógica, como **ATM - VC** (*Asynchronous Transfer Mode - Virtual Connection*) o virtual, como virtual **LAN/VLAN** [(*Local Area Network*)/(*Virtual Local Area Network*)], aunque otras referencias puedan ser utilizadas para la clasificación, como por ejemplo: direcciones **IP** de origen y destino, número de puerto de origen y de destino, octeto **ToS**, etc. Del resultado de la clasificación se obtiene los perfiles de tráfico, y las correspondientes normas para vigilancia, marcado y "*shaping*" de los paquetes de entrada.

- **Control, marcado y "*shaping*" en la entrada**

Para el servicio "garantizado": El control podría ser implementado con un algoritmo "*token bucket*", de modo que una parte de "*burst*" sea permitido. Cuando un paquete llega y hay *tokens* en el *bucket*, el paquete se considera un "perfil" y cuando no existe, el paquete se considera "fuera de perfil". El proceso de marcado pondrá, por ejemplo, el **DSCP** del paquete en el primero caso a 101.000, y en el segundo caso, a 100.000. El algoritmo **RED** (*Random Early Detection*) más tarde decide si desechará o pondrá el paquete en cola. Paquetes con **DSCP** diferente de 101.000 serán desechados antes de los paquetes con **DSCP** 100.000. Después de esto, todos los paquetes que no se descartan, sin importar si tiene perfil o no, se colocan en la misma cola para evitar su ejecución fuera de orden. Esta cola se llama de **AQ** (*Assured Queue*).

Para el servicio expreso: El control es normalmente implementado utilizando el algoritmo "*leaky bucket*" de manera que "*burst*" no es permitido. Cuando un paquete llega y tiene *tokens* en el *bucket*, el paquete es considerado como "correcto", y el **DSCP** del paquete se sitúa en 111.000. Si no hay *tokens* en el *bucket*, el paquete es considerado "incorrecto" y, según el **SLA** entre el **ISP** y el usuario, estos últimos paquetes pueden ser desechados inmediatamente, o pueden ser transmitidos y registrados pagando más. Si estos paquetes se encaminan a la transmisión, sus **DSCPs** también serán fijados en 111.000. Todos los paquetes se colocarán en la misma cola llamada **PQ** (*Premium Queue*).

- **Gestión de colas y "*scheduling*" de paquetes**

Una vez que el paquete está marcado, su tratamiento se determina sólo por su **DSCP**, sea este en el *router edge* o en el *router core*. En general, se mide de la manera siguiente:

- Para tráfico de paquetes *best effort* y *assured:* El algoritmo **RED** es inicialmente aplicado para comprobar si el paquete es el que será desechado. Si es así, esto se hará al azar, basado en su **DSCP** y en la duración media de la cola. Así, los paquetes de diferentes conexiones serán desechados de forma equivalente. Con esto, el mecanismo de control de flujo del **TCP** para estas conexiones consigue reducir su velocidad de transmisión en diferentes momentos.

Esto, a su vez, contribuye para evitar el *overflow* en las filas de los *routers*, evitando así el efecto *tail-drop*. En caso contrario, este efecto podría sincronizar la reducción de múltiples flujos **TCP** y, posteriormente, aumentar su velocidad de manera simultánea, causando la sobrecarga de tráfico y afectando significativamente el rendimiento de la red.

- **Para tráfico expreso:** Por lo general ni **RED** ni **RIO** (**RED** *with in and out*) se aplican, ya que no es conveniente descartar paquetes expresos. Todos los paquetes se colocan en la cola **PQ** (**RIO** es una forma de gestión de *buffer* más avanzada que **RED**). Todas las colas **PQ**, **AQ,** y a veces **DQ** (*Default Queue*), esta última de los paquetes *best effort*, son "programadas" (*scheduled*) por un algoritmo llamado **WFQ** (*Weight Fair Queueing*) o por una variante de él. La tasa de estas colas es definida por el administrador de la red sobre la base de estadísticas de tráfico y política del dominio. Si se define el factor de aprovisionamiento de una cola = pf (cola) como la relación entre la velocidad de salida configurada y la velocidad real de entrada, a continuación, el siguiente orden debe mantenerse:

$$pf\ (PF) > pf\ (AQ) > pf\ (DQ) > 1{,}0$$

- **La congestión:** Las situaciones de congestión en las redes **IP** son controladas principalmente por la determinación del orden que los paquetes serán enviados a la *interface* de salida y basadas en las prioridades asignadas a estos paquetes. Básicamente, existen 4 protocolos de encolamiento (*Queueing Protocols*), cada uno de los cuales permite la creación de diferentes números de colas, más o menos diferenciando el tráfico y especificando el orden en el que todo el tráfico será enviado. En períodos normales, es decir, sin congestión, los paquetes son enviados tan pronto como llegan. Durante la congestión de la *interface* de salida, los paquetes llegan más rápido que la *interface* puede enviarlos. Usando "gerenciamiento", los paquetes son puestos en cola hasta que la *interface* quede libre para enviarlos. En el ínterin los paquetes son preparados para la transmisión de acuerdo a su prioridad previamente establecida y a los mecanismos de gerenciamiento de colas configuradas en la *interface*. El *router* determina el orden de transmisión de paquetes, controlando cuales paquetes son asignados en cual

cola y como las colas son atendidas en relación a las otras. Los cuatro protocolos mencionados anteriormente son los siguientes:

FIFO (*First In First Out*): la transmisión está en orden en que llegan los paquetes.

WFQ: provee cola dinámica que divide el ancho de banda (*bandwidth*) entre las colas de tráfico basado en "pesos".

CQ (*Custom Queueing*): provee asignación de ancho de banda (*bandwidth*) proporcional a cada clase de tráfico.

PQ: los paquetes de una clase de prioridad se envían antes de los paquetes del tráfico de baja prioridad.

En los sistemas que no utilizan la gestión de congestión, **FIFO** se utiliza de forma predeterminada. En los sistemas de gestión, el **WFQ** es el más utilizado y es sensible al valor de "prioridad" configurado en el campo **DSCP** (sección 3.2). Esto permite detectar los paquetes de alta prioridad y la asignación de banda (*bandwidth*) a los flujos durante las situaciones de congestión en función de mayor o menor valor de este campo. **WFQ** prevé un peso para cada flujo que a su vez determina la secuencia de transmisión de paquetes en la cola.

En el caso, por ejemplo, de dos tráficos, uno de ellos con prioridad 7 y otro con 3, los pesos son diferentes e inversamente proporcionales a los valores de la prioridad y pesos menores se envían en primero lugar. El flujo de prioridad 7 y un peso menor serán enviados primero.

Para los *routers Cisco*, con **WFQ,** la determinación de la banda asignada es de acuerdo con el valor de prioridad en porcentaje de la banda total, procediendo de la siguiente manera:

$$\text{Banda asignada} = \frac{\text{Prioridad} + 1}{\sum n} \% \Rightarrow \quad A$$

Dónde: Prioridad = prioridad dada al flujo

n = "peso"

1 2 3 4 5 6 7 8 = n = "peso"

⬆⬆⬆⬆⬆⬆⬆⬆ (prioridad)

- **Ejemplo 1**: Ocho flujos de cada uno de ellos con una prioridad (0 a 7).
  Prioridad 0 recibe: 0 + 1/36 = 1/36 de ancho de banda.
  Prioridad 1 recibe: 1 + 1/36 = 2/36 de ancho de banda.
  Prioridad 7 recibe: 7 + 1 = 8/36 de ancho de banda.

- **Ejemplo 2**: Dos flujos en una banda "x". Uno con prioridad 0 y otro con prioridad variable (1 a 7). Utilizando la fórmula A, encontramos:

Cuadro 3.5: Prioridad IP

Prioridad	N	Banda para prioridad 0	Banda para prioridad variable	Porcentaje asignada a prioridad variable
0 y 1	1 + 2 = 3	1/3	2/3	0,666
0 y 2	1 + 3 = 4	1/4	3/4	0,750
0 y 3	1 + 4 = 5	1/5	4/5	0,800
0 y 4	1 + 5 = 6	1/6	5/6	0,833
0 y 5	1 + 6 = 7	1/7	6/7	0,857
0 y 6	1 + 7 = 8	1/8	7/8	0,875
0 y 7	1 + 8 = 9	1/9	8/9	0,888

d) MPLS

El término "*multilayer switching*" se utiliza generalmente para describir la integración de la conmutación de capa 2 con el enrutamiento de la capa 3. Los *switches* de la capa 2 proporcionan conectividad a grandes velocidades mientras que los *routers* **IP**, interconectados mediante mallas de circuitos virtuales de la capa 2, para proporcionar a la inteligencia para el encaminamiento de los datagramas **IP**. Actualmente, algunas redes de **ISPs** están construidas utilizando un modelo "*overlay*" en el cual una topología lógica de enrutamiento **IP** rueda sobre e independiente de la topología de la conmutación de la capa 2, del tipo **ATM** o *Frame Relay*. Sin embargo, las implementaciones con este enfoque tienen

varias dificultades, entre ellas la complejidad de la asignación de dos arquitecturas diferentes, lo que requiere, entre otras cosas, establecer y mantener las dos topologías, espacios de direcciones, protocolos de enrutamiento, protocolos de señalización y sistemas de asignación de recursos en forma aislada.

La evolución de *Internet* con el fin de reducir esta complejidad y la combinación de la conmutación de la capa 2 y el enrutamiento de la capa 3 en una solución totalmente integrada es todo acerca de las soluciones que salen de *multilayer switching*, dónde el **MPLS** es su último representante.

MPLS es una tecnología para el enrutamiento de paquetes que utiliza "etiquetas" para tomar decisiones de enrutamiento. Con ella, el análisis del encabezado de la capa 3 se hace una sola vez, cuando el paquete entra en el dominio **MPLS**. Luego, comprobar la etiqueta para el envío de paquetes subsecuentes.

Definida en la **RFC** 3031, **MPLS** proporciona una eficiente asignación, enrutamiento, encaminamiento y conmutación de los flujos de tráfico a través de una red. En general, desempeñará las siguientes funciones:

- Especifica los mecanismos para la gestión de los flujos de tráfico de diversas granularidades, tales como los flujos entre los distintos *hardwares*, máquinas o incluso flujos entre las diferentes aplicaciones.
- Sigue siendo independiente de los protocolos de capas 2 y 3.
- Proporciona un medio para asignar las direcciones **IP** con las etiquetas.
- Permite *interfaces* con los protocolos de enrutamiento existentes como **RSVP** y **OSPF** (*Open Shortest Path First*).
- Soporta los protocolos de nivel 2, **IP**, **ATM** y *Frame Relay*.

En esta tecnología, la transmisión de datos se produce en los **LSPs**, que son una secuencia de etiquetas para cada uno de los nodos a lo largo de la ruta entre origen y destino.

Las etiquetas que son identificadores específicos de un determinado flujo se distribuyen utilizando **LDP** (*Label Distribution Protocol*), **RSVP** o transmitidos en

los protocolos de enrutamiento como **BGP** o **OSPF**. Cada paquete de datos encapsula y lleva las etiquetas durante su viaje desde el origen al destino.

- **Los componentes de un dominio MPLS**

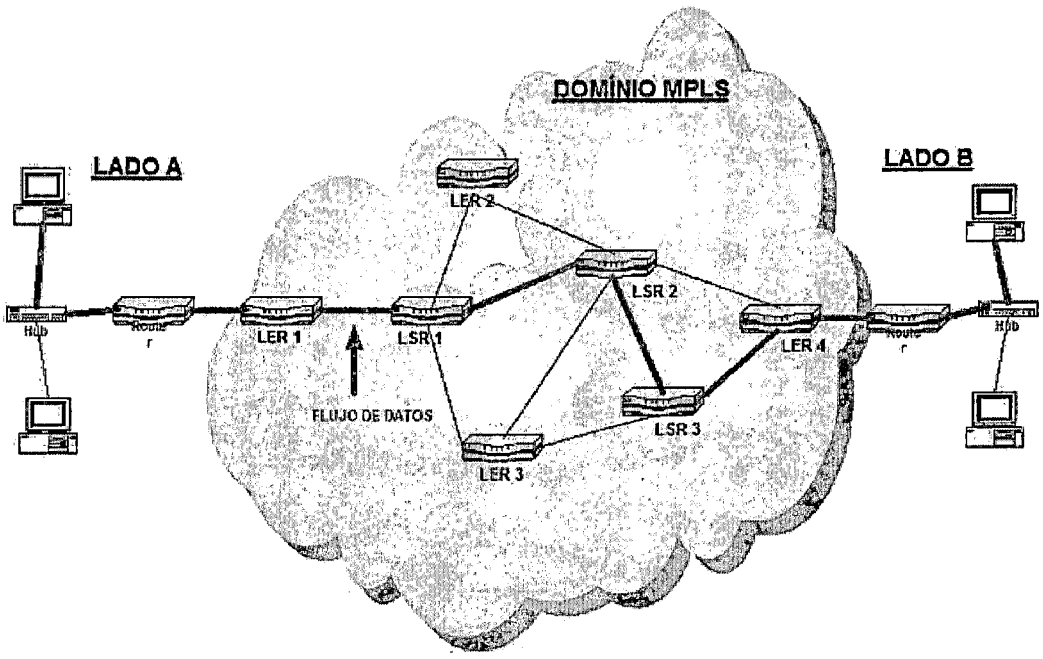
Los siguientes los componentes de un *backbone* **MPLS**:

- **LERs** (*Label Edge Routers*): son conocidos así los *routers* instalados en los extremos de la red de acceso a la red **MPLS**. Soportan múltiples puertos que pueden conectarse a distintas redes, como *Frame Relay*, **ATM**, *Ethernet*, etc. Encaminan los flujos de tráfico a la red **MPLS** después de haber establecido **LSPs** usando el protocolo de señalización de etiquetas en la entrada y distribuyen los flujos de nuevo a las redes de acceso en la salida. Representan el punto de entrada y salida de una red **MPLS**.

- **LSR** (*Label Switching Routers*): son los *routers* de alta velocidad instalados en el centro de una red **MPLS** y participar en el establecimiento de los **LSPs** usando los protocolos de señalización de etiquetas adecuadas.

- **FEC** (*Forward Equivalence Class*): es la representación de un grupo de paquetes que comparten los mismos requisitos para su transporte y tiene el mismo tratamiento de enrutamiento del origen al destino. Cada **LSR** construye un cuadro para especificar la forma en que el paquete debe ser enviado. En el **LIB** (*Label Information Base*), se incluyen los conjuntos de pares asociados de **FECs** - etiquetas.

Fig. 3.9: Dominio MPLS



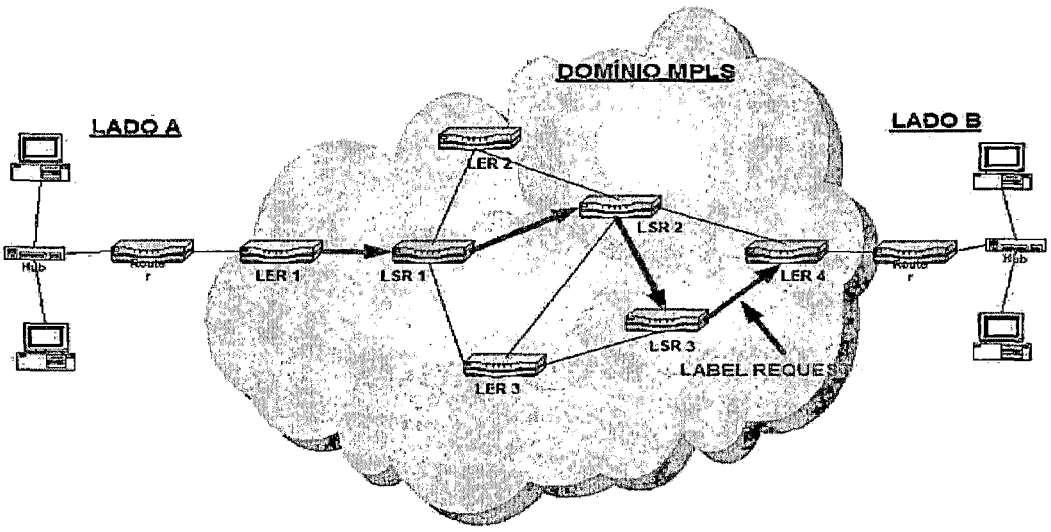
- **Actividades generadas en MPLS**

- **Creación de Etiquetas:** La creación de etiquetas se basa en el modelo "*Control Driven*". En este método, las etiquetas se crean cuando llega la información de control. Es decir, las etiquetas se asignan en respuesta al procesamiento normal del tráfico del protocolo de enrutamiento, al tráfico de control **RSVP** o en respuesta a una configuración estática. Los principales beneficios de este modelo son: las etiquetas son firmadas y distribuidas antes de la llegada del tráfico de datos del usuario. Esto significa que, si existe una ruta en el cuadro **IP**, una etiqueta ya se ha asignado a esta ruta, de manera que el tráfico, llegado al *router*, pueda ser cambiado inmediatamente. La escalabilidad es mejorada por el hecho de que el número de **LSPs** esta en proporción al número de entradas en la cuadro de enrutamiento y no el número de flujos individuales. En este modelo, se establecen los **LSPs** sólo después de un cambio de topología o la llegada de un control de tráfico y no con la llegada de cada nuevo flujo de tráfico. Cada paquete en el flujo se cambia de etiqueta no sólo al fin del flujo, como en el caso del modelo *data-driven*.

- **Distribución de Etiquetas:** El **IETF** ha definido un nuevo protocolo, conocido como **LDP** para la señalización explícita y la gestión de las etiquetas. Versiones

con extensiones al protocolo básico **LDP** también se han definido para soportar el enrutamiento explícito basado en requisitos de **QoS** y **CoS**, como por ejemplo, el protocolo de **CR-LDP** (*Constraint Routing - Label Distribution Protocol*). La arquitectura **MPLS** no define un único método para la señalización de la distribución de etiquetas. Algunos protocolos, como **BGP**, se han mejorado para incluir las etiquetas dentro del "*frame*" del protocolo. Otros, como el protocolo **RSVP**, también se han mejorado para servir a este fin (**RSVP - TE**). En general, podemos decir que el **LDP** es ampliamente utilizado para el asignamiento de destinos **IP** en las etiquetas, el **RSVP - TE** y **CR-LDP** para ingeniería de tráfico y reserva de los recursos, el **PIM** (*Protocol Independent Multicast*) para estados *multicast* de asignamiento de las etiquetas, el **BGP** para las etiquetas externas (**VPNs**), etc.

Fig. 3.10: Solicitud de Etiquetas



- **LSPs:** Un *path* es un camino definido para el envío de un determinado flujo basado en una **FEC**. **MPLS** proporciona las siguientes dos opciones para la definición del **LSP**:

- ***Hop by Hop Routing:*** En esta opción, cada **LSR**, sin tener en cuenta, selecciona el siguiente *link* para un determinado **FEC**, similar de las redes **IP** convencionales, utilizando los protocolos de enrutamiento convencionales, como

OSPF, **PNNI** (*Private Network-to-Network Interface*), **IS-IS** (*Intermediate System to Intermediate System*), etc.

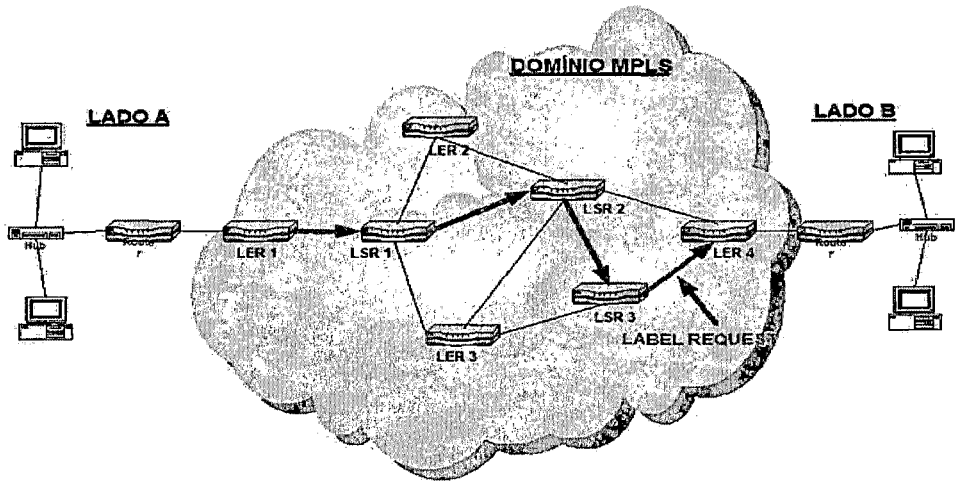
- ***Explicit Routing:*** En este caso, el **LSR** de entrada especifica la lista de nodos (**LSRs**) a través de los cuales viajara. La ruta especificada puede no ser la mejor o la más corta, ya que lo que importa es la reserva de recursos para el cumplimiento de los requisitos de **QoS** solicitados.

El **LSP** es unidireccional. Para el tráfico de retorno es utilizado como **LSP**. La etiqueta (*label*) identifica el camino (*path*) que el paquete debe recorrer.

En su entrada, en el dominio **MPLS**, el paquete está clasificado como nuevo o perteneciendo a una determinada **FEC**. Una etiqueta se ha asignado al flujo y el valor de la etiqueta es normalmente derivado, en lo general el protocolo de capa 2 que se está utilizando. Para *Frame Relay*, la **DLCIs** (*Data Link Control Identifiers*), y para **ATM** los **VPIs/VCIs** (*Virtual Path Identifiers/Virtual Channel Identifiers*) pueden ser utilizados directamente como etiquetas. Tales etiquetas son, después, asociadas a una **FEC** como resultado de cualquier acto o política por defecto. Esta asociación etiqueta - **FEC** se suele hacer de dos maneras:

- Por plataforma: los valores son únicos en el **LSR**. Las etiquetas se asignan a partir de un único *pool*, de modo que no se distribuirán dos etiquetas del mismo valor.
- Por *interface*: los valores son asociados con las *interfaces* y pueden tener el mismo valor en diferentes *interfaces*.

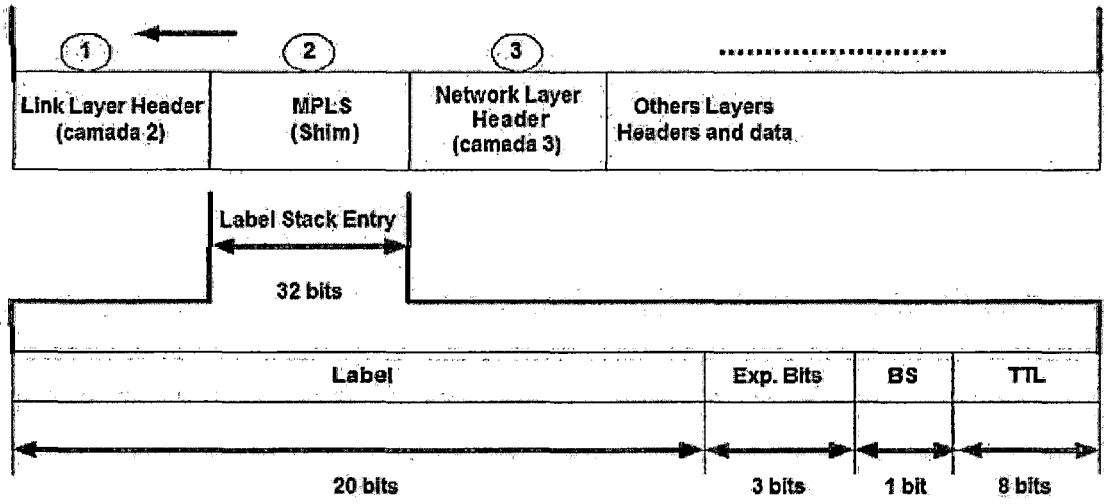
Fig. 3.11: Creación de los LSPs



e) Estructura de las Etiquetas

Una etiqueta es un identificador con significado local de corta duración, de longitud fija de 4 *bytes*, que es utilizado para identificar una **FEC**. La etiqueta que es puesta en un determinado paquete representa la **FEC** dónde el paquete es asignado.

Fig. 3.12: Estructura genérica de la etiqueta MPLS



Un conjunto de *label stack entries* constituye un "*label stack*".

Label: valor de la etiqueta - 20 *bits*.
Exp (*experimental use)* - 3 *bits* (utilizado como campo para **CoS**).
BS (*Bottom of Stack*) - 1 *bit*.
TTL (*Time to live*) - 8 *bits*.

La etiqueta puede ser incluida en el encabezado de la capa 2 o entre el encabezado de la capa 2 y capa 3 (*Shim*).

El rango de valores de etiquetas puede ser de 0 a 2^{20}-1, o 1.048.575 etiquetas. De acuerdo al **RFC** 3032 - **MPLS** *label stack encoding*, las etiquetas de valores de 0 a 15 son reservadas con el siguiente significado:

- Etiquetas de valores de 4 a 15 son reservadas para uso futuro. Los valores de 0 a 3 son utilizados de la siguiente manera:

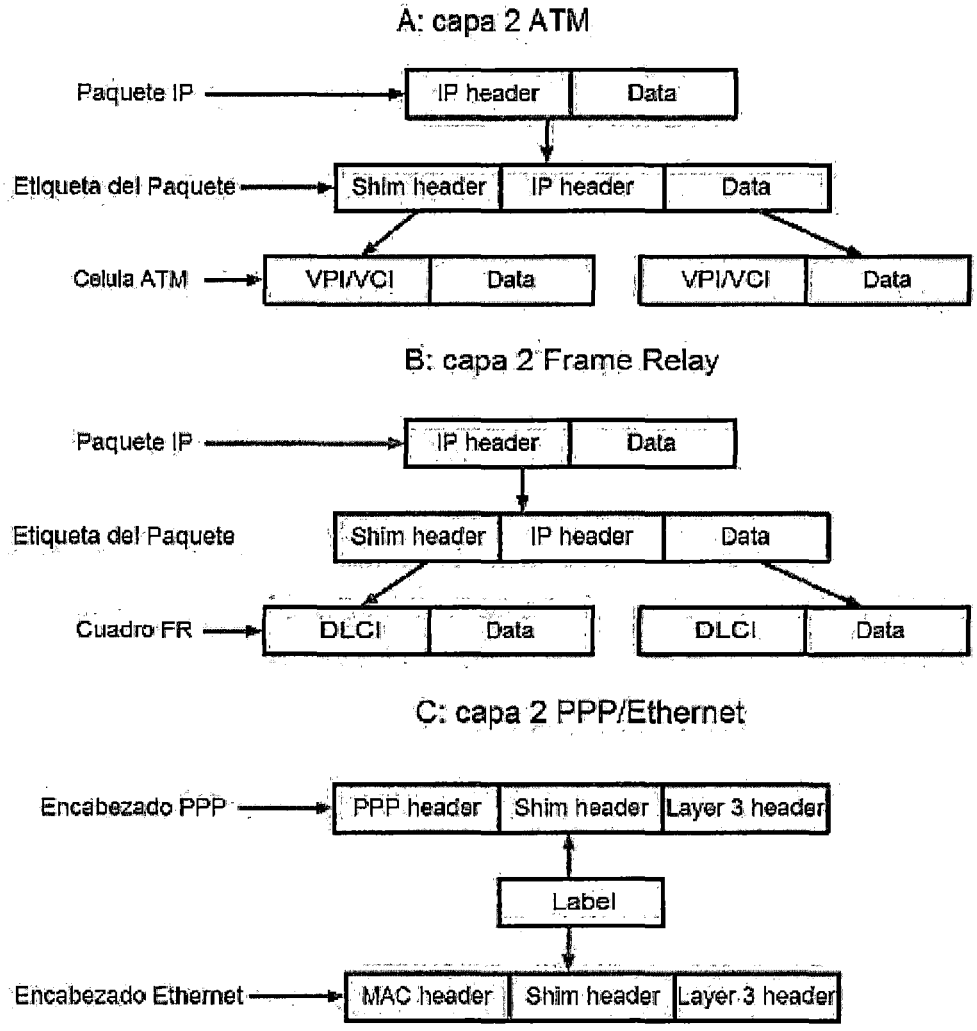
Valor 0 representa el "**IPv4** *explicit null label*" e indica que la ruta del paquete debe estar basada en el encabezado **IPv4**.

Valor 1 representa el "*router alert label*". Cuando un paquete recibido contiene esa etiqueta en el inicio del "*label stack*", será encaminado para un módulo de *software* para su procesamiento. El correcto encaminamiento del paquete será determinado por la etiqueta siguiente en el "*label stack*". Su uso es análogo al "*router alert option*" en paquetes **IP**.

Valor 2 representa el "**IPv6** *explicit null label*", lo que indica que el paquete debe utilizar el encabezado **IPv6**.

Valor 3 representa el "*implicit null label*", etiqueta que un **LSR** puede asignar y distribuir, pero sin nunca aparecer en el encapsulamiento. Esto indica que un **LSR** saca la etiqueta de entrada del *stack* y encamina el restante del paquete (con la etiqueta o no) a través de la *interface* de salida (según con la entrada en el **LIB**). Aunque este valor puede no figurar en el encapsulamiento, el necesita ser especificado en el **LDP** (por lo tanto, este valor es reservado).

Fig. 3.13: Posicionamiento de la etiqueta



f) Label Distribution Protocol

El **LDP** es un nuevo protocolo especificado en la **RFC** 3036 para la distribución de etiquetas en redes **MPLS**. Es utilizado para asignamiento de etiquetas a la **FECs**, las cuales, a su vez, crean **LSPs**. Las sesiones del **LDP** son establecidas entre las capas del **LDP** no necesariamente adyacentes. Entre estas capas se intercambiará:

- *Discovery Messages:* que anuncian y mantienen la presencia de un **LSR** en la red.
- *Session Messages:* que establecen, mantienen y terminan las sesiones entre las capas del **LDP**.

- *Advertisement Messages:* que crean, cambian y eliminan el asignamiento de las etiquetas - **FECs**.
- *Notification Messages:* que proporcionan informaciones de alerta y los errores.

En el **MPLS**, la distribución de las etiquetas puede ocurrir de dos maneras: *downstream on demand* y *downstream* no solicitada. En el primero, un **LSR** distribuye un conjunto **FEC** - etiqueta en respuesta a una solicitud explícita de otro **LSR**. En la segunda, es una respuesta a una solicitud no explícita de **LSRs**. Ambas pueden ser utilizadas en la misma red al mismo tiempo.

Además, el control de la distribución depende del modo de operar del **LSR**: con el control "independiente" o con el control "ordenado" (*ordered*). Con control independiente, el **LSR** puede hacer el asignamiento de las etiquetas a sus vecinos en cualquier momento. Por ejemplo, cuando el modo de funcionamiento es independiente/*downstream on demand*, un **LSR** puede responder las llamadas de asignamiento de etiquetas inmediatamente, sin esperar por asignamiento del *link* cercano. En caso contrario, tendría que esperar hasta estar dispuesta a conmutar la etiqueta de esa **FEC**. Una consecuencia de esto es que, usando el modo independiente, una etiqueta *upstream* puede ser anunciada antes que una etiqueta *downstream* sea recibida.

En el modo *ordered*, el **LSR** podrá iniciar la transmisión de una asignación de etiqueta sólo para una **FEC**, la cual tiene asignado en el próximo *link*, o para aquella **FEC** donde el **LSR** es la salida. El **LSR** debe esperar hasta que una etiqueta de un **LSR** *downstream* sea recibido antes de poder asignar una **FEC** y pasar las correspondientes etiquetas a los **LSRs** *upstream*.
Este protocolo no permite implementar Ingeniería de Tráfico.

El formato general del mensaje **LDP** se indica en el siguiente cuadro:

Cuadro 3.6: Formato general de mensaje LDP

16 bits		16 bits
U	Message Type	Message Length
Message ID		
Mandatory Parameters		
Optional Parameters		

U = *bit* de mensaje desconocido. Si U = 0, un mensaje de notificación vuelve al transmisor del mensaje. Si U = 1, el mensaje es considerado "desconocido" e ignorado.

Message Type es el tipo de mensaje.

Message Length, indica el tamaño total del mensaje, compuesto de *Message* **ID** (*Internet Draft*) + *mandatory parameters* + *optional parameters in octetos*.

Message **ID** con tamaño de 32 *bits* es utilizado para identificar el mensaje. Es utilizado por el **LSR**, que está enviando el mensaje para facilitar la identificación de mensajes de notificación relacionadas con este mensaje.

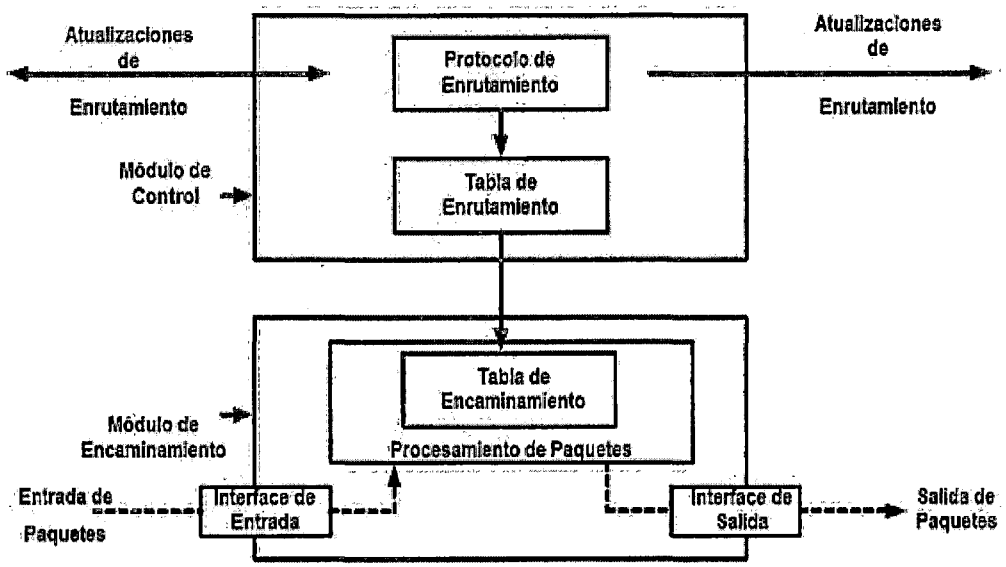
Mandatory Parameters, es el conjunto de parámetros requeridos por la mayoría de los mensajes, cuyo longitud es variable (algunos mensajes, no requieren estos parámetros).

Optional Parameters, es el conjunto de parámetros opcionales de la mayoría de los mensajes. Su longitud es variable (algunos mensajes no tienen parámetros variables). Para mensajes con parámetros variables, ellos pueden aparecer en cualquier orden.

g) Operación del MPLS

La solución **MPLS** es compuesta de dos módulos funcionales, un módulo de control y un módulo de encaminamiento (Fig. 3.14). El módulo de control utiliza protocolos de enrutamiento *standards*, como **OSPF**, **IS-IS** y **BGP**-4, para el intercambio de informaciones con otros *routers,* construyendo y manteniendo el cuadro de enrutamiento. El módulo de enrutamiento es basado en el algoritmo de intercambio y enrutamiento de etiquetas, igual que el utilizado para encaminar los datos en los *switches* **ATM** y *frame relay*. Actualmente hay protocolos para este fin, entre los cuales podemos mencionar **LDP**, **CR_LDP**, **RSVP**, y **BGP**. La **RFC** 3031, no específica que protocolo debe ser utilizado, ya que depende de las convocatorias que deben ser atendidas por la red privada.

Fig. 3.14: Módulo de la solución MPLS



- **Módulo de Control**

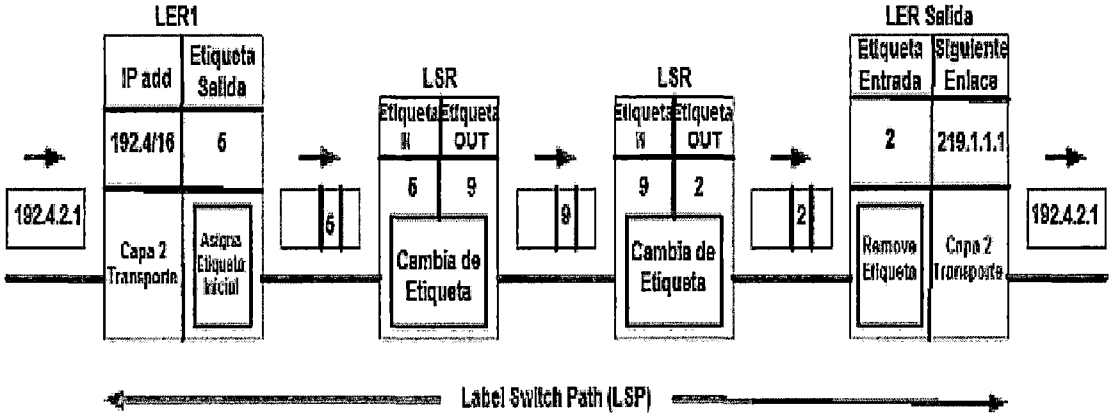
Cuando llegan los paquetes en la entrada del *router*, el módulo de enrutamiento realiza una búsqueda en el cuadro de enrutamiento mantenida por el módulo de control para tomar decisiones de enrutamiento de cada paquete. Examina, en concreto, la información contenida en el encabezado del paquete, la búsqueda por la cuadro de enrutamiento de información que sea igual del encabezado del paquete y dirige los paquetes de la *interface* de entrada para la *interface* de salida

correspondiente. En la Fig. 3.15, muestra el proceso dónde el **LER1** recibe un paquete a la dirección **IP** 192.4.2.1. El **LER1** busca en el cuadro y observa una **FEC** 192.4/16, hace el asignamiento del paquete a esa **FEC**, pone la etiqueta con valor 5 y remite al siguiente *link* a través del **LSP**.

- **Módulo de Enrutamiento**

En el núcleo de la red, los **LSRs** ignoran las informaciones del encabezado del paquete y simplemente encaminan el paquete utilizando el algoritmo de cambio de etiquetas. Cuando el paquete etiquetado llega al **LSR**, el módulo de enrutamiento utiliza el número de puerto de entrada y la etiqueta de búsqueda en su cuadro de enrutamiento. Cuando encuentra una información relacionada con ellos, saca de ella los datos de la etiqueta de salida, *interface* de salida y la dirección del *link* siguiente. Luego, cambia la etiqueta de entrada con la etiqueta de salida y dirige el paquete a la *interface* de salida, visando la transmisión al siguiente *link*, a través del **LSP**.

Fig. 3.15: Operación del MPLS



Cuando el paquete etiquetado llega al **LSR** de salida, el módulo de enrutamiento también hace una búsqueda en su cuadro de enrutamiento. Si el *link* siguiente no es un **LSR**, desecha la etiqueta y envía el paquete utilizando enrutamiento **IP** convencional.

h) MPLS e Ingeniería de Tráfico (TE)

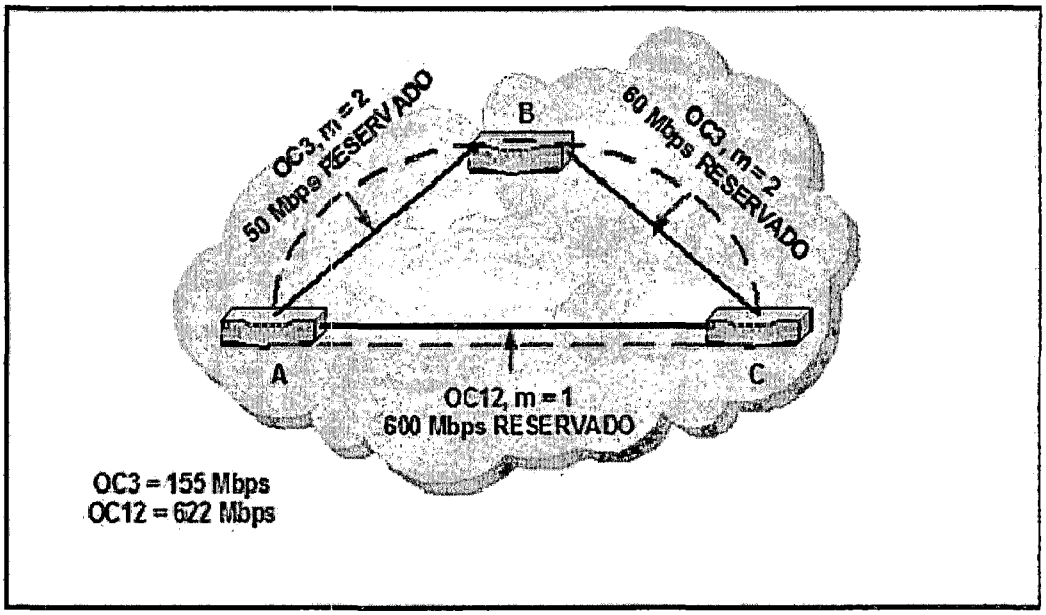
En las redes **IP** convencionales, por lo general son utilizados los protocolos "*shortest path*" para el encaminamiento de los flujos de tráfico, lo que puede causar situaciones de congestión en algunos *links*, mientras que puede causar menor utilización de otros *links*. La propuesta del **TE** es precisamente la optimización de las características y el rendimiento de la red. Para esto es necesario tener un cierto control sobre las rutas de los paquetes creados por **MPLS**. Con **TE**, el enrutamiento es basado en "restricciones" (*constraint routing*), técnicas que consideran no sólo los requisitos para **QoS** si no también otros parámetros para la elección de la ruta. Esta no será necesariamente la más corta, sin embargo la mejor en la política previamente definida, como por ejemplo, una ruta menos congestionada. El principal protocolo es el **CR_LDP**, detallado a continuación junto con otros parámetros de Ingeniería de Tráfico.

Pueden también ser utilizados algunos protocolos convencionales, como el **OSPF**, por ejemplo. Sin embargo, deberían ser compatibles con **CR** y, por tanto, deben tener la ampliación **TE** (**OSPF - TE**) para su aplicación (abarcan informaciones no incluidas en el modelo convencional, como ancho de banda, prioridad, máximo ancho de banda reservable, etc.)

- ***Constraint Route***

Se puede traducir *constraint route* como "enrutamiento basado en restricciones". Este enrutamiento lleva además de las consideraciones para determinación de las rutas, otros parámetros además de la topología de la red. Puede seleccionar un vínculo más distante, pero ligeramente cargado, en vez de un vínculo cercano, pero muy cargado. Por ejemplo, en la Fig. 3.16, es necesario un *enlace* de 40 Mbps entre A y C:

Fig. 3.16: *Constraint Routing*



El *enlace* más corto entre el *router* A y C es directamente a través del enlace A-C, con medida de **IGP**, m = 1. Debido al ancho de banda reservable en el vínculo más corto debe ser sólo 22 Mbps (622-600), cuando el enrutamiento **CR** intenta encontrar el mejor *link* para un **LSP** de 40 Mbps, selecciona el enlace A-B-C en lugar del A-C. Esta elección es debido a que el enlace más corto no puede atender la "restricción" de ancho de banda. Cabe señalar que el ancho de banda reservable en un vínculo es igual al máximo reservable, establecido por el administrador, menos el total reservado para el **LSP** en el vínculo, no dependiendo de la cantidad real disponible en el *enlace*. Por ejemplo, si el máximo reservable en el *link* es 155 Mbps, y el total máximo reservado para el **LSP** es 100 Mbps, el máximo reservable en el *link* es de 55 Mbps, no teniendo en cuenta si el *link* está o no transporta 100 Mbps de tráfico. En otras palabras, el enrutamiento **CR** no calcula los **LSPs** basado en el valor instantáneo residual de ancho de banda de los *links*.

- **El protocolo CR-LDP**

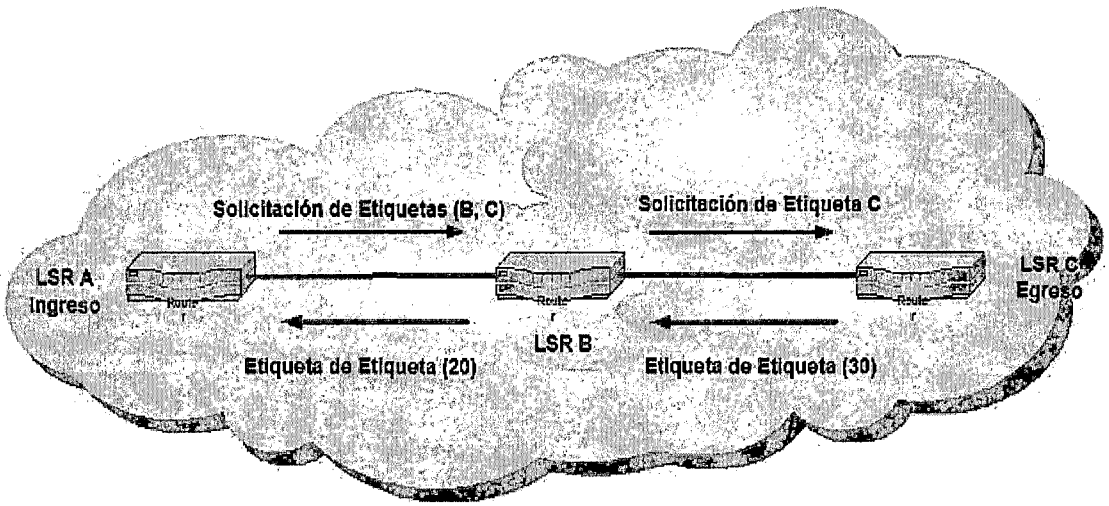
Como se mencionó anteriormente, el **MPLS** dispone de mecanismos para atender a las dos estructuras propuestas para la prestación de **QoS**. La primera utiliza

extensiones **TE** al protocolo de señalización **RSVP** para asistir al **IntServ** y la segunda, denominada **CR-LDP**, utiliza extensión al **LDP** para asistir al **DiffServ**.

Tal vez la principal herramienta ofrecida por el **MPLS** es la capacidad de configurar rutas de **CR-LSP** con las restricciones de Ingeniería de Tráfico asociadas a ellas.

El **CR-LDP** es un protocolo de señalización para el establecimiento de **LSPs** "explícitos" unidireccionales con atributos de **QoS** anexados en flujos de tráfico entrante en un dominio **MPLS**. A continuación, se describe brevemente el procedimiento de operación del **CR-LDP** (Fig. 3.17).

Fig. 3.17: Establecimiento de LSPs con CR-LDP



El *router* de entrada (**LSR** A), después de determinar la necesidad de establecer un nuevo **LSP** hasta el **LSR** C, envía un mensaje de "solicitación de etiqueta" (*label request*) con ruta explícita B, C para el *router* **LSR** B.

El *router* **LSR** B recibe el mensaje de solicitación de etiqueta, verifica que no sea el *router* de "salida" para aquel **LSP** y encamina el mensaje hacia adelante a lo largo de la ruta indicada en el propio mensaje. Reserva los recursos solicitados para el nuevo **LSP** y envía el mensaje para el **LSR** C.

El **LSR** C señala que es la "salida" para este nuevo **LSP**. Efectúa, entonces, la reserva de recursos solicitada, elije una etiqueta y la envía de vuelta para **LSR** B en un mensaje de "asignamiento de la etiqueta" (*label mapping*) que contiene los parámetros finales de reserva del nuevo **LSP**.

El **LSR** B recibe el "*label mapping*" y verifica su correspondencia con el mensaje de solicitación de etiqueta a través del identificador de **LSP** (**LSP ID**), que es el mismo para ambos mensajes. Termina la reserva de recursos, elige una etiqueta para este nuevo **LSP** y envía para el **LSR** A en un mensaje de asignamiento de la etiqueta.

El **LSR** A se comporta de forma similar como ocurre con el **LSR** B, pero sin la opción de etiqueta, porque él es el *router* de "entrada".

Los formatos de los diversos mensajes intercambiados entre los **LSRs** y los campos incluidos en ellos se muestran a continuación.

- **El formato del mensaje "solicitación de etiqueta"**

Cuadro 3.7: Mensaje "solicitación de etiqueta"

16 bits	16 bits
Label Request (0 x 401)	Message Length
Message ID	
FEC TLV	Obligatorio
LSPID TLV	Opcional
ER-TLV	Opcional
Traffic TLV	Opcional
Pinning TLV	Opcional
Resource Class TLV	Opcional
Pre-emption TLV	Opcional

FEC TLV es igual al **FEC** tipo **CR-LSP** (*Constraint-Based Routing Label Switched Path*) (0 x 04): es utilizada únicamente en los mensajes **CR-LSPs**. **LSPID TLV** (*Label Switched Path Internet Draft Time, Length and Value*), es un identificador único de un **CR-LSP** dentro de la red **MPLS**. Está compuesto del *ingress* **LSR** *router* **ID** (o una de sus propias direcciones **IPv4**) y un único **CR-LSP ID** de este **LSR**.

ER TLV (*Explicit Routing Time, Length and Value*), es un mensaje que especifica la ruta que ha de adoptar el **LSP** que está siendo establecido.

Traffic **TLV,** es utilizado para señalar los valores de los parámetros de tráfico necesarios por un **CR-LSP**.

Pinning **TLV**, permite asegurar que, una vez que la reserva de recursos se ha establecido sobre una ruta, el *router* no reemplazará esta ruta por una mejor (a menos que la ruta establecida no sea válida).

Resource Class **TLV,** también conocida como *color* **TLV**, es utilizada para especificar cuáles *links* son aceptables por el **CR-LSP**.

Pre-emption **TLV**, es cuando una ruta con los recursos suficientes no puede ser encontrada, los **LSPs** existentes pueden ser redirigidos para reasignar recursos para el nuevo **LSP**. En este tema son configurados los valores para este caso.

- **El formato del campo *traffic* TLV del mensaje "solicitación de etiqueta"**

Se muestra a continuación el contenido del campo **TLV.**

Cuadro 3.8: Campo TLV de la "solicitación de etiqueta"

16 bits			16 bits
0	0	Type	Length
Flags	Frequency	Reserved	Weight
Peak Data Rate (PDR)			
Peak Burst Size (PBS)			
Committed Data Rate (CDR)			
Committed Burst Size (CBS)			
Excess Burst Size (EBS)			

Valores 0/0, son parte de la codificación general **TLV** y llamados de *bits* "e" y "f". Tiene que ver con el tratamiento y encaminamiento del mensaje cuando el campo "*type*", a continuación, no es reconocido.

Type es el campo de 14 *bits* que informa el valor de los parámetros de tráfico **TLV** de valor 0 x 0810.

Length, especifica la longitud del campo con un valor de 24 *bytes*.

Flags es el campo de 8 *bits* con la siguiente especificación:

F8 F7 F6 F5 F4 F3 F2 F1.
F8 y F7 son reservados. Cero en la transmisión e ignorados en la recepción.
F6, corresponde al *weight*.
F5, corresponde al **EBS** (*Excess Burst Size*).
F4, corresponde al **CBS** (*Committed Burst Size*).
F3, corresponde al **CDR** (*Committed Burst Rate*).
F2, corresponde al **PBS** (*Peak Burst Size*).
F1, corresponde al **PRD** (*Peak Data Rate*).

Para cada valor de F1, con valor 0 indica no negociable, si es 1 indica negociable.

Frequency, es un campo de 8 *bits* con el siguiente significado:

0 = no especificado.
1 = frecuente.
2 = muy frecuente.
3-255 = reservado (0 en la transmisión e ignorado en la recepción).

La frecuencia especifica a que granularidad la **CDR** asignada ese **CR-LSP** tiene disponibilidad. Un valor "muy frecuente" significa que la velocidad disponible debe ser, el menor cuando se miden durante cualquier intervalo de tiempo igual o mayor que el tiempo del paquete más pequeño en el **CDR**. El valor "frecuente" significa lo mismo que el anterior, pero medido sobre un intervalo de tiempo igual o mayor que un pequeño número de tiempos de paquetes más pequeños en el **CDR**.

Weight es el campo de 8 *bits* que indica el "peso" del **CR-LSP**. Los valores válidos son de 1 a 255, cero significa que el peso no es aplicable a ese **CR-LSP**.

Traffic Parameters, **PDR**, **PBS**, **CDR**, **CBS**, **EBS** son codificados en 32 *bits*. **PDR/CDR** en *bytes* por segundo y **PBS**, **CBS** y **EBS** en *bytes*.

PDR, especifica la máxima velocidad en la cual el tráfico deberá ser enviado al **CR-LSP**.

PBS es el tamaño máximo de una porcion en el **PDR**.

CDR es la velocidad que el campo **MPLS** garantiza y se compromete a mantener en el **CR-LSP**.

CBS es el tamaño máximo de una porcion en el **CDR**.

EBS es el tamaño del tráfico enviado en un **CR-LSP** que supere el *committed rate*.

- **El formato del mensaje "asignado de etiqueta"**

Cuadro 3.9: Formato del mensaje "asignado de etiqueta"

16 Bits		16 Bits
0	Label Mapping (0 x 400)	Message Length
Message ID		
FEC TLV		
Label TLV		
Label Request Message ID TLV		
LSPID TLV		Opcional
Traffic TLV		Opcional

Un mensaje es enviado por un **LSR** *downstream* a un **LSR** *upstream* en una de las siguientes condiciones.

- Cuando el **LSR** es la "salida" del **CR-LSP** y fue solicitado un asignamiento *upstream*.
- Cuando el **LSR** ha recibido una asignación de su **LSR** *downstream* para un **CR-LSP**, una solicitud *upstream* sigue pendiente.

i) Aplicaciones de MPLS

Actualmente, las aplicaciones más comunes del **MPLS** en las redes **IP** son las siguientes:

- Redes Privadas Virtuales (**VPNs**).
- Ingeniería de Tráfico (**TE**).
- Clase de Servicio (**CoS**).

- **Redes Privadas Virtuales (VPNs)**

Una **VPN** simula la operación de una **WAN** (*Wide Area Network*) privada sobre la red pública *Internet*. Para prestar un servicio viable de **VPN**, los **ISPs** deben

resolver, entre otras cosas, los problemas de privacidad de los datos y soportar el uso de direcciones **IP** privadas; no solo dentro de una **VPN**. **MPLS** proporciona una solución sencilla y eficaz a ambos los desafíos en la toma de las decisiones de enrutamiento basadas en el valor de la etiqueta y no en la dirección de destino del encabezado del paquete.

- **Ingeniería de Tráfico (TE)**

La Ingeniería de Tráfico permite mover los flujos de tráfico por el camino más corto calculado por el **IGP** (*Interior Gateway Protocol*), es decir, por caminos físicos potencialmente menos congestionados a través de la red. Esta es, actualmente, la principal aplicación del **MPLS** debido, principalmente, a la enorme demanda de recursos de la red y la naturaleza *mission-critical* de las aplicaciones **IP**. **MPLS** es muy adecuado para proporcionar la base para permitir **TE** en grandes redes, principalmente por las siguientes razones:

- El soporte para caminos explícitos permite especificar la ruta física exacta que un **LSP** toma a través de la red.
- Estadísticas por **LSP** pueden ser utilizadas como entradas para la planificación de redes y herramientas de análisis para identificar embotellamientos y usos de los trazos, y para planificar expansiones futuras.
- Enrutamiento basado en restricciones ofrece funciones mejoradas que permiten a los **LSPs** atender los requisitos específicos de funcionamiento.
- Soluciones basadas en **MPLS** pueden ejecutarse sobre redes orientadas a paquetes y no sólo limitadas a la infraestructura **ATM**.
- Además, la Ingeniería de Tráfico es también muy importante en la provisión de **QoS** en la *Internet* debido a su propósito básico de optimización de las características y rendimiento de la red en condiciones normales. Eso ya no ocurre en el servicio **DiffServ**, que proporciona degradación diferenciada del rendimiento de los diferentes flujos de tráfico no sólo durante las situaciones de congestión de la red, sino que, sin la congestión, el rendimiento de la red se mantiene igual, como si este servicio no estuviera presente.

- **Clase de Servicio (CoS)**

MPLS se puede utilizar para implementar principalmente Servicio Diferenciado o **DiffServ** como es mas conocida. Evidentemente, eso sólo será posible en conjunto con recursos adicionales de **TE** y técnicas de clasificación de tráfico. Dos enfoques se pueden utilizar:

- El tráfico que fluye a través de un **LSP** puede ser encolado para transmisión en la *interface* de salida de cada **LSR** basado en la configuración de los *bits* de prioridad del encabezado **MPLS**.
- Se puede esperar múltiples **LSPs** entre cada par de **LERs**. Cada **LSP** puede ser configurado con **TE** para proporcionar diferente rendimiento y garantías de ancho de banda. Se puede poner tráfico de alta prioridad en un **LSP**, de prioridad mediana en otro tráfico, *best effort* en un tercero y, si fuera el caso, tráfico inferior que *best effort* en un cuarto.

Los *bits* de prioridad son utilizados sólo para ordenar los paquetes en una de muchas clases de servicio. Por lo general es el **ISP** que provee el servicio específico soportado por cada clasificación de servicio.

j) Un ejemplo de aplicación de MPLS

La Fig. 3.18 ilustra la operación del **MPLS** en relación al enrutamiento convencional de paquetes **IP**. La operación muestra el enrutamiento mejorado al soportar aplicaciones que requieren algo más que el enrutamiento basado en la dirección de destino.

- **Enrutamiento Convencional**

Suponiendo que los *routers* funcionan inicialmente con enrutamiento convencional de paquetes **IP**, si la computadora A o B pasa un paquete a la computadora C, entonces el paquete continuara el camino 1 a través del núcleo de la red, debido, a este as el camino más corto calculado por el **IGP**.

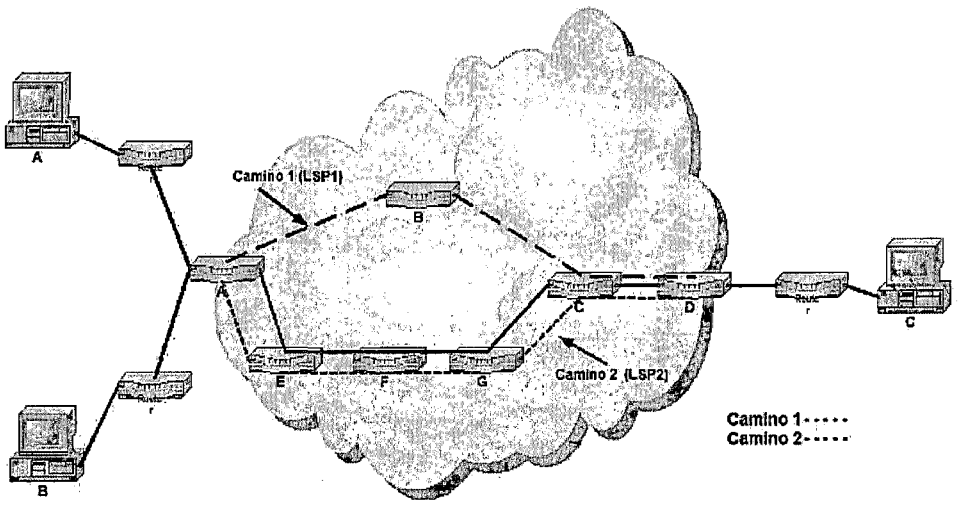
Suponiendo que es necesario implementar una política para controlar la congestión en el *router* B, una forma de reducir la congestión en la B podría ser la

distribución de la carga del tráfico sobre los diferentes caminos en la red. El tráfico originado en la computadora A y enviado a computadora C podría seguir el camino más corto, o ruta 1. El tráfico originado en la B y destinado a C podría seguir en el otro sentido, el 2. Con enrutamiento **IP** convencional, esta política no podría ser implementada porque todo el encaminamiento de tráfico del *router* A está basado en la dirección del paquete de destino (ruta 1).

- **Enrutamiento con MPLS**

Si los *routers* en el núcleo de la red funcionan como **LSRs**, es más fácil implementar una política para reducir la congestión en el *router* B. Se puede, por ejemplo, configurar el **LSP**1 para seguir el camino 1 y el **LSP**2 para seguir el camino 2. Además de esto, se puede configurar el *router* **LSR** para que transmita todo el tráfico recibido desde la computadora A y destinado a C, en el **LSP**1 y, del mismo modo, todo el tráfico de la computadora B es destinado a C, en el **LSP**2. De esta forma, **MPLS** permite atribuir cualquier **FEC** a un **LSP** construido por el usuario con un control preciso de tráfico fluyendo en la red.

Fig. 3.18: Ejemplo de aplicación



k) Servicio DiffServ con MPLS

MPLS y **DiffServ** etiquetan los paquetes después de clasificarlos. **MPLS**, con las llamadas "etiquetas **MPLS**", y **DiffServ**, con los llamados **DSCPs**.

Los **DSCPs**, básicamente, establecen la prioridad y prioridad de rechazo de los paquetes. Las etiquetas **MPLS** determinan el camino que el paquete seguirá. Pueden ser utilizados recursos de **TE** para la atribución de algunas etiquetas a caminos con algunas características. En general, combinando los recursos de ambas técnicas, es posible, por ejemplo, especificar las rutas que los paquetes deberán seguir y su comportamiento en las colas de los diferentes *routers*.

Los *routers* **MPLS** no examinan el encabezado **IP** de los paquetes, pero el **DSCP** de los datagramas **IP** está contenido en el encabezado **IP**. Así que, para usar **DiffServ** en redes **MPLS**, la información de **DiffServ** contenida en el encabezado **IP** debe ser asignada a la etiqueta **MPLS** del paquete. Las formas más utilizadas actualmente son: **E-LSP** (**EXP**-*Inferred*-**LSP**) y **L-LSP** (*Label-Only-Inferred*-**LSP**).

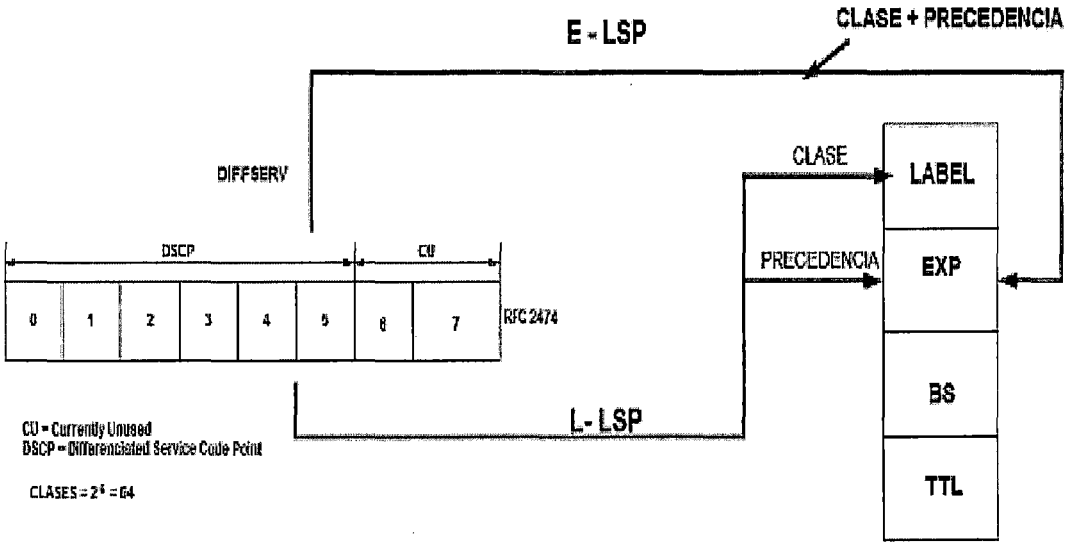
E-LSP: se utiliza cuando múltiples **BAs** (*Behavior Aggregate*) son asignados al mismo **LSP**. Todos los paquetes tendrán la misma etiqueta y el campo **EXP** (*expected*) será utilizado para especificar el **PHB** aplicable a cada paquete. Este **PHB** incluye parámetros de *scheduling* y preferencia de rechazo.

L-LSP: si sólo un **BA** asignado a un único **LSP**, el **DSCP** es codificado implícitamente en la etiqueta, pudiendo el campo **EXP** ser utilizado para codificar la preferencia de rechazo de los paquetes. Este método resuelve el problema de los campos cortos para codificación del **DSCP**, tales como **E-LSP**. Sin embargo eso impone altos requerimientos al sistema. El número de etiquetas posible crece y, de la misma forma, la cantidad de recursos necesarios en la red, lo que puede provocar serios problemas de escalabilidad.

Cuadro 3.10: Asignamiento E-LSP

Clase			Prioridad					PHB	Decimal	Prioridad
0	1	2	3	4	5	6	7			
0	0	0	0	0	0	0	0	best effort	0	
0	0	1	0	1	0	0	0	AF11	10	baja
0	0	1	1	0	0	0	0	AF12	12	mediana
0	0	1	1	1	0	0	0	AF13	14	alta
0	1	0	0	1	0	0	0	AF21	18	baja
0	1	0	1	0	0	0	0	AF22	20	mediana
0	1	0	1	1	0	0	0	AF23	22	alta
0	1	1	0	1	0	0	0	AF31	26	baja
0	1	1	1	0	0	0	0	AF32	28	mediana
0	1	1	1	1	0	0	0	AF33	30	alta
1	0	0	0	1	0	0	0	AF41	34	baja
1	0	0	1	0	0	0	0	AF42	36	mediana
1	0	0	1	1	0	0	0	AF43	38	alta
1	0	1	1	1	0	0	0	Expedited	46	

Fig. 3.19: Modelos de mapeamiento



- **DiffServ con MPLS asignado al campo EXP (E-LSP)**

Una implementación de **DiffServ** con **MPLS** utilizando el campo **EXP** podría resumirse así:

- En el **LSR** de entrada, es insertado en el paquete un encabezado **MPLS**. El **DSCP** en el encabezado **IP** es asignado al campo experimental (**EXP** *field*) del encabezado de **MPLS**.
- Es hecha la clasificación **BA** basada en el **EXP** *field* en vez del **DSCP**.
- En el **LSR** de salida, es eliminado el encabezado **MPLS**.

Los paquetes mantienen sus **DSCPs** en la entrada de la red. Cuando los encabezados **MPLS** son insertados en el **LSR** de entrada, los **DSCPs** son asignados en el campo **EXP** del encabezado **MPLS**. Porque el **DSCP** tiene 6 *bits* de tamaño y el campo **EXP** sólo 3, alguna información se puede perder en este proceso. Sin embargo, sólo los 3 *bits* más a la izquierda (los 3 *bits* de la antigua prioridad) del **DSCP** contienen información útil y se pueden copiar en el campo **EXP**. Por lo tanto, los 2 *bits* más a la izquierda indican la clase de servicio y, por tanto, la cola. El tercer *bit* indica cuando el paquete está dentro del perfil o cuando hay probabilidad de rechazo.

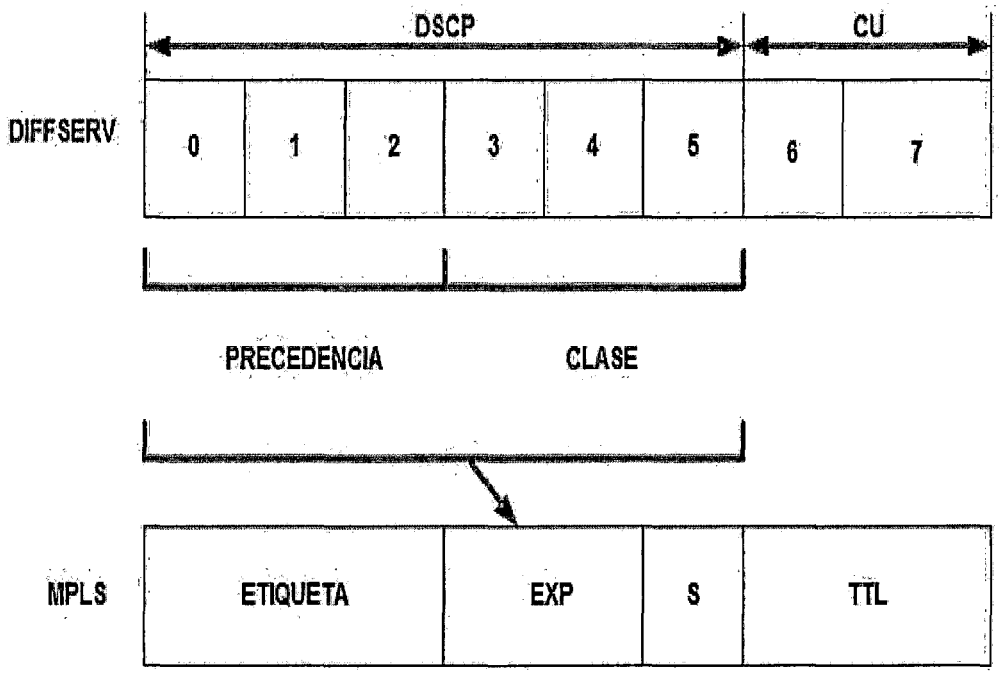
Cuadro 3.11: Mapeamiento DSCP –EXP

Servicio	Perfil Adecuado	Fuera de Perfil	Desecho
Premium	111	111	Ningún
Asegurado	101	100	Paquetes 100

En el medio de un **LSP**, la clasificación **BA** es basada en el campo **EXP** en vez del **DSCP**. Las gestiones de *buffers* y *scheduling* son idénticas con y sin **MPLS**. Por lo tanto, el **PHB** de los paquetes también es idéntico, que es deseable por significar que:

- Los **ISPs** que proporcionan **QoS** con **MPLS** pueden fácilmente interoperar con **ISPs** que proporcionan **QoS** sin **MPLS**.
- El papel del **MPLS** en la provisión de **QoS** es transparente para varios usuarios finales.

Fig. 3.20: Mapeamiento DSCP - E-LSP



* Normalmente son copiados los bits 3, 4 y 5 al campo EXP

* O definindo una politica de mapeo

* Mas utilizado actualmente

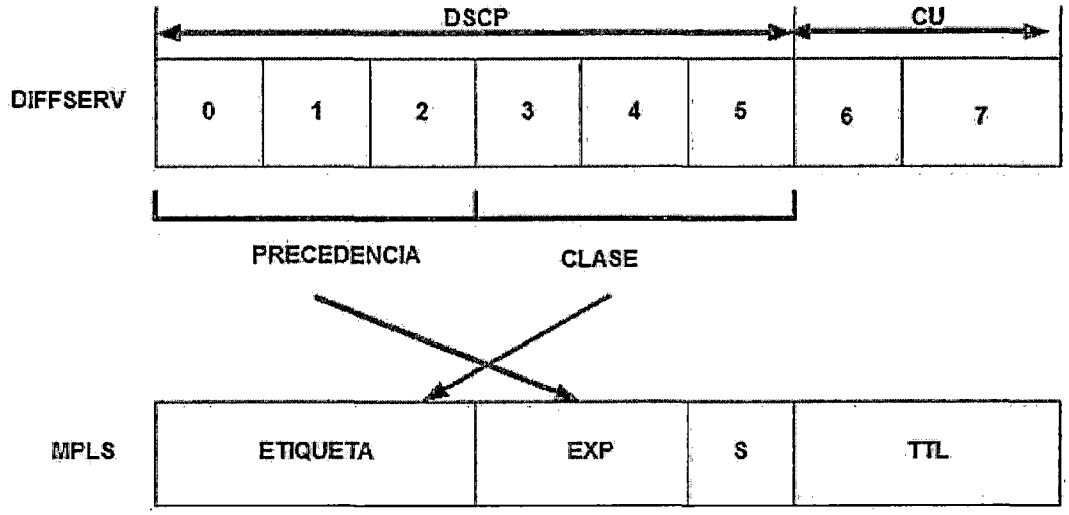
Cuadro 3.12: Ejemplo de asignamiento E-LSP

Class	DSCP		EXP	
Reserved for control plane traffic	Class selector 7	111000	7	111
Reserved for control plane traffic	Class selector 6	110000	6	110
Class 1 (real-time traffic)	EF	101000	5	101
Class 2 in-profile	AF31	100000	4	100
Class 2 out-of-profile	AF32, AF33	011000	3	011
Class 3 in-profile	AF11	010000	2	010
Class 3 out-of-profile	AF12, AF13	001000	1	001
Class 4 (best effort)	Default	000000	0	000

- **DiffServ con MPLS asignado a la etiqueta y campo EXP (L-LSP)**

El **L-LSP** y un **LSP** dónde los nodos de la red infieren en el tratamiento de **QoS** para los paquetes **MPLS** de la etiqueta y del campo **EXP** del paquete **MPLS**. En general de la etiqueta es derivada a la clase y del **EXP** a prioridad de rechazo. ¿Cuál es mejor **E-LSP** o **L-LSP**? Depende del diseño de **QoS**. Para **ATM** y *frame relay* el **L-LSP** es la única alternativa. Si se necesita de gran número de clases y prioridad de rechazo, entonces el **L-LSP** es necesario, aunque los **ISPs** que actualmente implantan servicios de **QoS** utilizan en el máximo 4 clases. En este caso **E-LSP** reúne los requisitos.

Fig. 3.21: Asignamiento DSCP - L-LSP



3.3.3 Evaluación del rendimiento de las técnicas de implementación de QOS.

De acuerdo con los conceptos presentados anteriormente, observamos lo siguiente: **QoS** representa, en general, el conjunto de técnicas necesarias para gestionar los parámetros de tráfico transportado o *throughput*, retraso o latencia, *jitter* (variación de retraso) y pérdida de paquetes (*packet loss*). **MPLS**, en su forma básica, es un protocolo que proporciona sólo una conmutación súper-rápida de paquetes en relación a los protocolos convencionales. El gran mérito está en las características de **TE**, **CoS** y **VPN**, que pueden ser utilizados sólo o en conjunto para varios fines, como implementación de **QoS**, en este caso.

El propósito principal de la Ingeniería de Tráfico es la reducción de los efectos de congestión en la red y, en la práctica, esto se traduce en la posibilidad de construcción de "caminos explícitos", recurso no disponible con encaminamiento convencional. Los recursos de **CoS** tienen por objetivo una diferenciación entre clases de servicio, y en la práctica, implican en la construcción de "caminos explícitos" y "reserva de recursos" (ancho de banda, por ejemplo). La reserva de banda puede ser hecha implícitamente (definida a través de *bandwidth*) o explícitamente a través de la configuración del parámetro de prioridad. Con estos conceptos como prioridad, se busca en la investigación, mostrar la efectividad de

los recursos de **TE** y **CoS** del **MPLS** en una plataforma de pruebas experimentales, montada específicamente para este fin e ilustrar su impacto en el rendimiento a través de mediciones comparativas de tráfico transportado y retraso en relación a la misma situación sin estos recursos. El Cuadro 3.13 se muestra un resumen de los recursos y posibilidades del **MPLS** en lo que se refiere a la implementación de **QoS**.

Cuadro 3.13: Resumen de recursos MPLS

RECURSO	IP convencional	IP con MPLS básico	MPLS + TE	MPLS + TE + CoS
Caminos Explícitos	No	No	Si	Si
Reserva de Recursos	No	No	Si	Si
Clases de Servicios	No	No	No	Si

a) Parámetros representativos de QoS

En la práctica, **QoS** implica el control de los parámetros de *throughput*, retraso, *jitter*, pérdida de paquetes, entre otros. A continuación una breve descripción de estos parámetros.

- **Tráfico Transportado (*throughput*)**

Considerado el parámetro más importante de **QoS**, tráfico transportado es una medida de la cantidad de datos que son encaminados en un cierto intervalo de tiempo. Normalmente es expresado en *bits* por segundo. La forma como las medidas son hechas afecta los resultados y sólo es posible comparar valores efectuados en las mismas condiciones. Por ejemplo, dos *routers* diferentes, con diferentes *throughputs*, no tiene mucho significado, a menos que, entre otras cosas, sean indicadas las características del tamaño de paquete de prueba y el ambiente de prueba para cada uno de ellos.

- **Retraso (*delay*)**

Retraso es simplemente la cantidad de tiempo que los paquetes utilizan para cruzar la red analizada. Puede ser medido de dos formas, en un sólo sentido o en el tiempo que lleva de ida y vuelta **RTT** (*Round Trip Time*). Mediciones de retraso en un sólo sentido son más sofisticadas y complicadas que **RTT**. Normalmente se mide el **RTT** y el resultado se divide por 2. **RTT** también es conocido como retraso fin-a-fin y representa la acumulación de los retrasos de transmisión y colas en los *routers*, retrasos de propagación y retrasos en los sistemas terminales a lo largo de una ruta del origen al destino. En esta investigación se consideró el **RTT** de ida y vuelta, en todas las mediciones de retraso. De la misma forma que con el *throughput*, la forma que es medido afecta los resultados. Pueden ser logrados valores más chicos de retraso cuando la medida se inicia en el instante que se recibe el último octeto del paquete y finalizase la medida cuando se transmite el primero octeto. Lo ideal sería medir desde el momento que el primero o el último octeto son recibidos y finalizar cuando el primero o el último octeto son transmitidos. Igualmente el tamaño del paquete puede hacer diferencia para las medidas de retraso. En el caso de un *router* necesita copiar un paquete de datos dentro de su propia memoria, un paquete grande usará más tiempo para copiar que un paquete pequeño. Conociéndose como las medidas fueron obtenidas (cuando el paquete empieza y termina), y el tamaño de los paquetes, podremos convertir dos medidas diferentes a una base común para fines de comparación.

Para una llamada telefónica local el **RTT**, tiempo de ida y vuelta de un paquete debe ser inferior de 150ms. Para el caso de llamadas internacionales se puede admitir hasta 1 segundo.

- **Variación de retraso (Jitter)**

Es definido como la fluctuación del tiempo transcurrido de paquete a paquete desde que es generado en la fuente hasta cuando es recibido en el destino, debido a las variaciones del retraso dentro de una red. Por ejemplo, para transmisión de audio y video no tienen mucha diferencia si los paquetes llevan 20ms o 30ms para que sean entregados, eso sucederá sólo cuando el tiempo de encaminamiento se mantenga constante, es decir, sean 20 o 30 siempre. Pero si

algunos paquetes llevan 20 y otros 30 para que sean entregados, esto ocasionará una calidad de sonido o imagen diferente (degradación).

Para **QoS** normalmente se define: "99% de los paquetes serán entregados con un retraso de 24,5 a 25,5ms", siendo estos valores verdaderos, posibles de conseguir y que llevan en cuenta ciertos niveles de congestión.

- **Pérdida de Paquetes (*Packet loss*)**

Puede ser definida como la fracción del total de paquetes enviados que no llegan a su destino. Puede representar un problema serio para aplicaciones de voz, dónde la pérdida de partes de voz digitalizada implica en la pérdida de calidad eventualmente no acepta por la aplicación. Las pérdidas de paquetes normalmente son consecuencia de:

Descarte de paquetes en los *routers*.

Pérdidas de paquetes debido a errores ocurridos en la línea.

Valores de pérdidas entre 1% y 5%, pueden ser tolerados dependiendo de cómo la voz fue codificada y transmitida, y de como la pérdida es tratada en el receptor.

b) Experiencias Efectuadas

Las experiencias efectuadas consistirán en el montaje, configuración y medición de los parámetros más representativos de **QoS**, tráfico transportado y retraso. Tráfico transportado o equivalentemente, ancho de banda, debido a que es un parámetro que todas las aplicaciones precisan y por tanto el más básico y presente en las especificaciones de **QoS**. El retraso de forma similar para las aplicaciones multimedia de audio conferencia, por ejemplo, dónde no es suficiente asegurar banda solamente, y los retrasos de comunicación y pérdida de paquetes influencian la interactividad de los usuarios y en la calidad de la aplicación. Las experiencias fueron efectuadas en los siguientes escenarios:

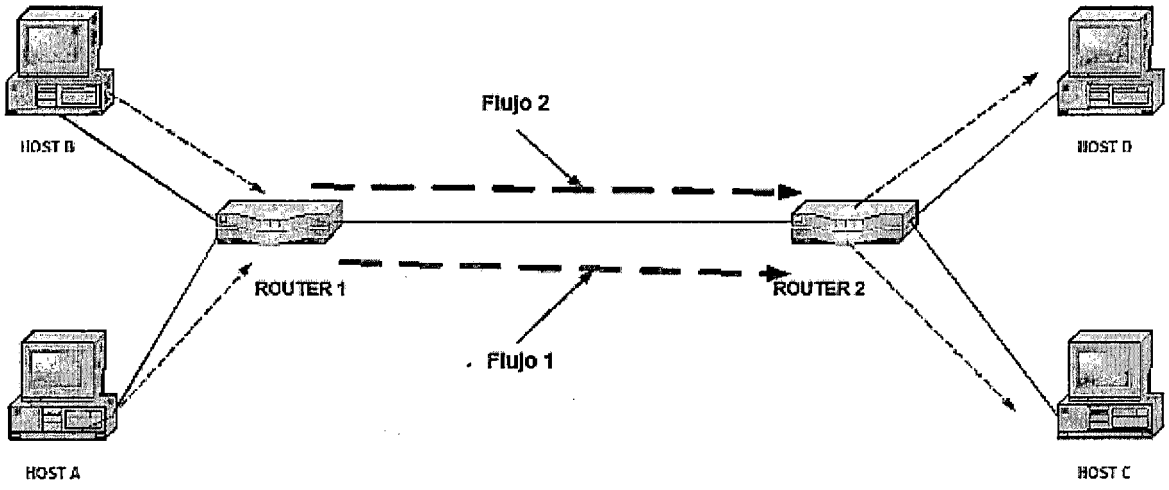
1. **IP** convencional.
2. **MPLS** básico.
3. Caminos explícitos (**MPLS** + **TE**).
4. Clase de servicio (**MPLS** + **CoS**).

Los efectos de la aplicación de los recursos de **MPLS** son visualizados en estas experiencias a través de los gráficos resultantes de las mediciones de tráfico transportado y retraso referenciados siempre en la misma situación en análisis, pero sin los recursos indicados. Estas mediciones no pretenden que sean exactas ni completas. La idea es observar que utilizando determinado recurso del MPLS se obtiene una mejora del rendimiento, sin preocuparnos por el valor de esta mejora. Para estas pruebas fueron utilizados los *softwares Netperf* e *Iperf*, con sus características detalladas en la sección 4.3.4 así como el comando ping del Windows. También fue utilizado el instrumento "Domino *Internetwork Analyzer*" de la *Wandel & Golterman* para la evaluación comparativa de algunas mediciones de los *softwares* mencionados.

- **Experiencia 1: IP convencional**

Esta experiencia fue efectuada sobre la plataforma de la Fig. 3.22 entre los *routers* 1 y 2, con el objetivo de obtener las referencias de *throughput* y retraso para las experiencias posteriores. Sobre esta plataforma fueron efectuadas las mediciones de referencia para uno y dos flujos simultáneos cruzando el vínculo de 2 Mbps. Los *routers* no incluyen ningún recurso adicional de **MPLS** o equivalente.

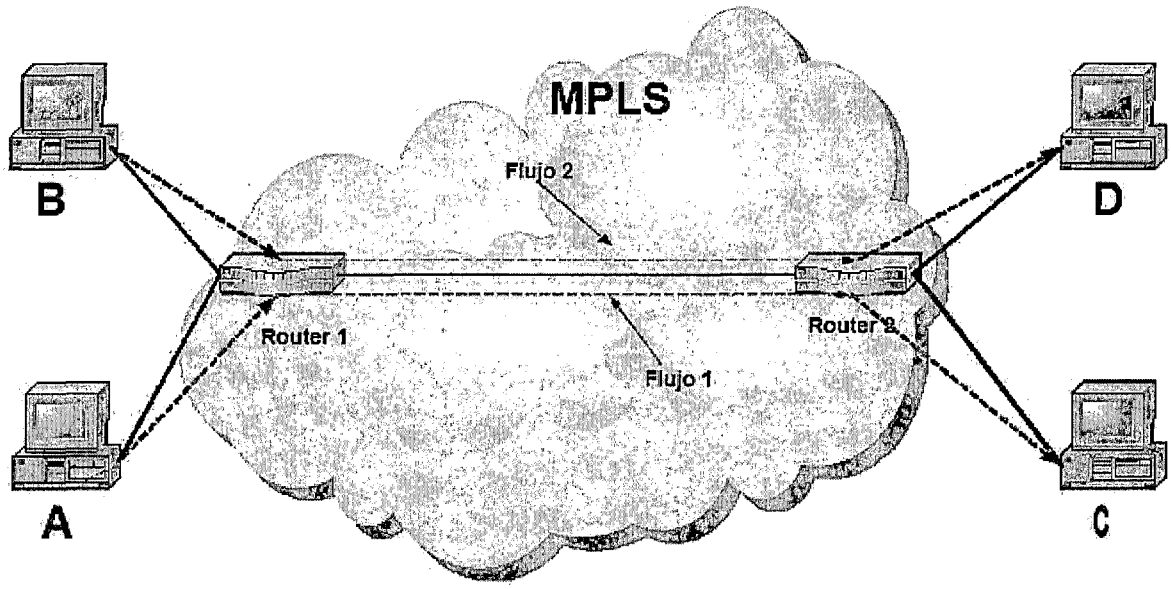
Fig. 3.22: Experiencia 1: IP convencional



- **Experiencia 2: MPLS básico**

Esta experiencia fue efectuada sobre la plataforma de la Fig. 3.23, entre los *routers* 1 y 2, con el objetivo de mostrar que solamente los recursos de **MPLS** básico habilitados en los *routers* (sobre el *frame relay* y **OSPF** de la sección anterior) no aseguran ninguna mejora de rendimiento.

Fig.3.23: MPLS básico



- **Experiencia 3: Caminos explícitos (*explicit* LSPs)**

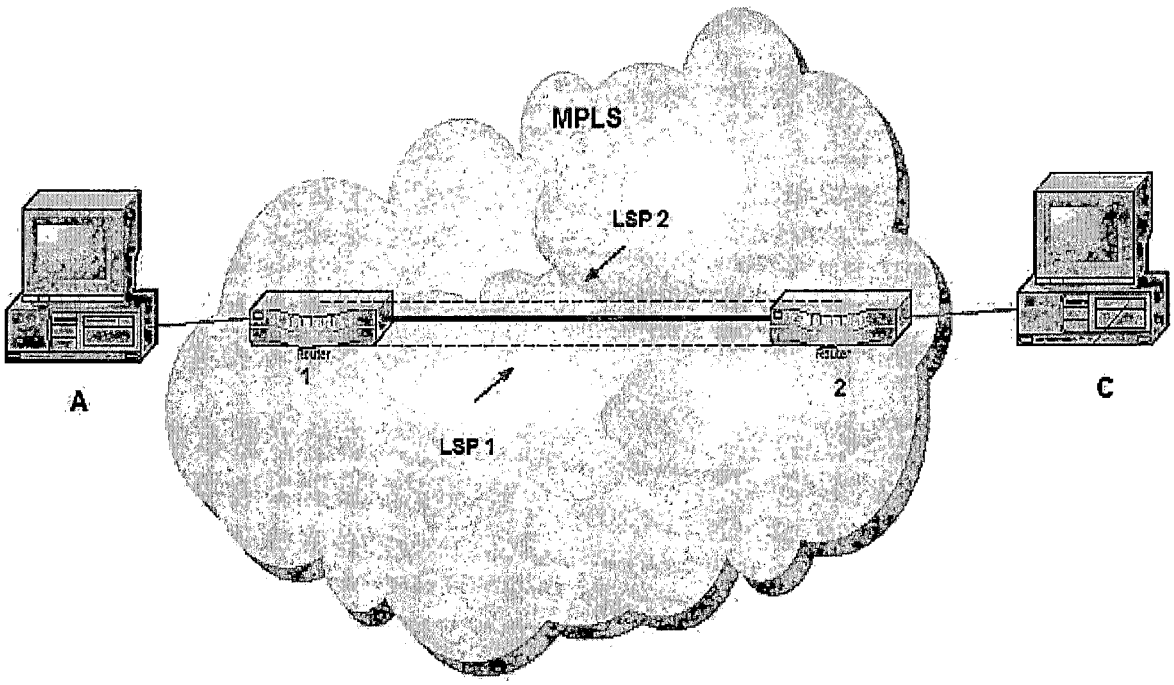
El objetivo es mostrar como los recursos de Ingeniería de Tráfico del **MPLS** pueden mejorar el rendimiento de una red en las situaciones de congestión. Para esto, aplicamos encaminamiento explícito para construir dos "caminos explícitos" en la plataforma montada. El procedimiento seguido considera tres situaciones. La primera con un flujo y sin **TE**, la segunda con dos flujos sin **TE** y la tercera con dos flujos con **TE**.

- **Dos caminos explícitos y un flujo de tráfico:** El objetivo es mostrar que sin **TE**, con caminos explícitos **MPLS**, no se consigue mejora de rendimiento para un flujo en el vínculo.

1. Se establece dos "caminos explícitos", **LSP**1 y **LSP**2.

2. Utilizando el *software Netperf* en la computadora A como cliente y en el otro lado la computadora C como servidor, medimos tráfico transportado y retraso para esta situación (1 único flujo en el vínculo).

Fig. 3.24: Caminos explícitos, un flujo

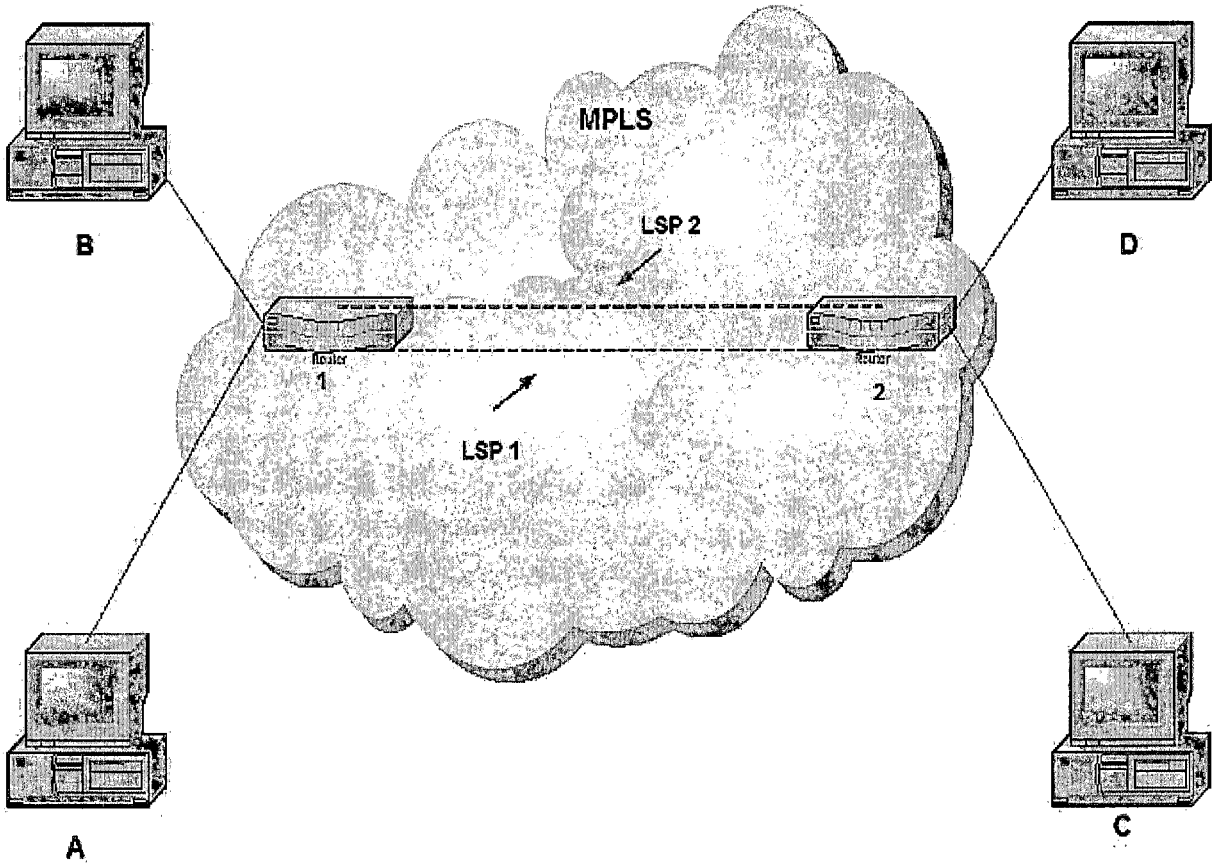


- **Dos caminos explícitos, dos flujos de tráfico:** El objetivo es mostrar que sin **TE**, con los mismos caminos explícitos **MPLS**, no se consigue mejorar el rendimiento para los dos flujos simultáneos en el vínculo.

1. Sigue siendo la situación de la sección anterior para A - C y el procedimiento es el mismo con la computadora B como cliente y D como servidor.

2. Se mide tráfico transportado y retraso para esta nueva situación.

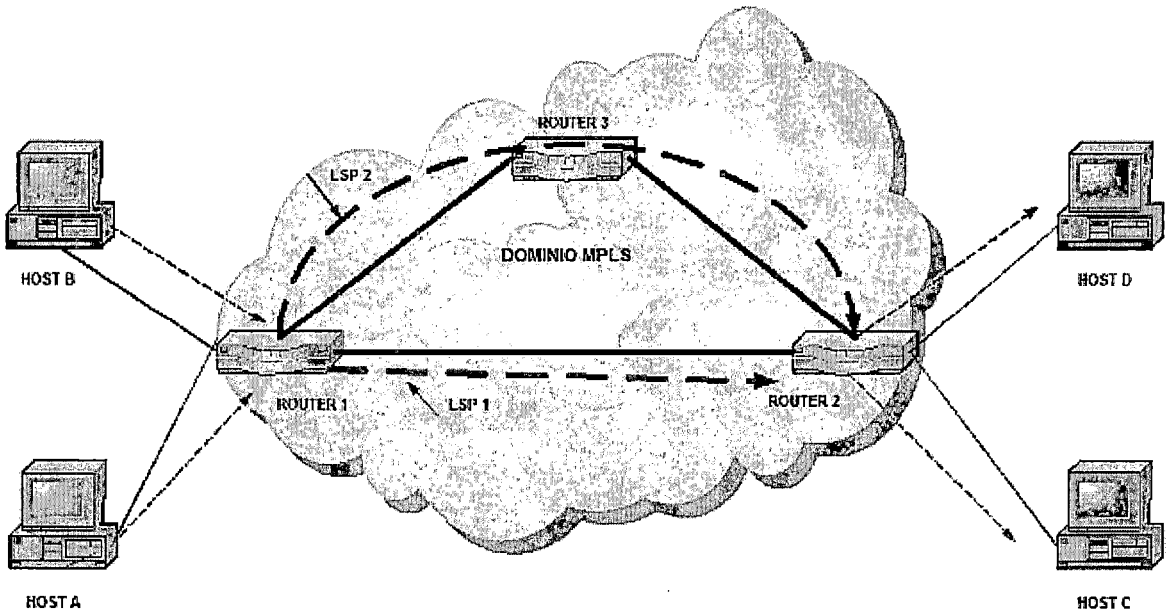
Fig. 3.25: Caminos explícitos, dos flujos



- **Dos caminos explícitos, dos flujos diversificados:** El objetivo es mostrar como utilizando **TE** podemos efectivamente mejorar el rendimiento de una red. El procedimiento seguido es el siguiente:

1. Desviamos el **LSP**2 para pasar a través del *router* 3, manteniendo el **LSP**1 en la ruta más corta (*router* 1 - *router* 2).
2. Se mide el tráfico transportado y retraso (1 flujo en el vínculo directo y uno en el vínculo a través del *router* 3).

Fig. 3.26: Caminos explícitos, dos flujos diversificados



- **Experiencia 4: Clases de servicio (MPLS + CoS)**

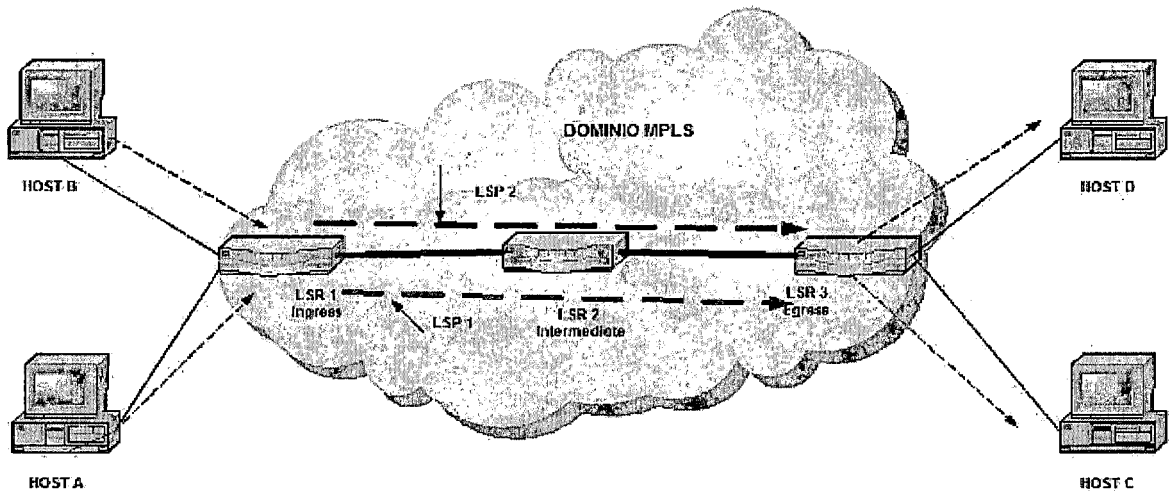
La experiencia 4 tiene como objetivo mostrar la efectividad de la reserva de recursos del **CoS** en general y del ancho de banda en particular, en sus dos formas básicas, explícita (a través del comando *bandwidth*) y a través de la configuración de la prioridad **IP.** Consta de:

- **Reserva de ancho de banda:** El objetivo es mostrar la efectividad de la reserva de ancho de banda explícita cuando es utilizado recursos de **CoS** del **MPLS** y comparado los valores medidos con los valores configurados. Explícita quieres decir "explícitamente configurada en los *routers*" a través del comando "*Bandwidth*". La Fig. 3.27 muestra la topología de red utilizada para esta experiencia. A cada flujo de tráfico de entrada es asignada una clase en función de las *interfaces Ethernet* de entrada (ETH0/0 = clase 1, ETH0/1 = clase 2) y cada *router* está configurado con reserva de ancho de banda. El procedimiento seguido es el siguiente:

1. Se establecen dos **LSPs**: **LSP**1 y **LSP**2.
2. Se establece reserva de ancho de banda en los **LSPs**.
3. 70% de la banda disponible (2 Mbps) para **LSP**1 (clase 1).
4. 30% restantes para **LSP**2 (clase 2).
5. Se asigna el **LSP**1 al tráfico de A para C y el **LSP**2 al tráfico de B para D.

La prueba implica la configuración del *router* **LSR**1, (*router* de entrada), para que clasifique y haga el asignamiento de los flujos de entrada para **LSP**1 y **LSP**2, en la entrada del dominio **MPLS** basados en la *interface* de entrada. El mismo para los *routers* intermediarios y de salida. Serán construidos gráficos que muestren la banda efectivamente asignada para cada uno de los flujos.

Fig. 3.27: CoS, ancho de banda/prioridad IP



- **Prioridad IP:** El objetivo es mostrar los efectos de la configuración de la prioridad en la división de la banda sobre nuestra red experimental y comparar los resultados de las mediciones efectuadas con los cálculos teóricos Serán construidos gráficos que muestren las bandas asignadas en función de la prioridad configurada para cada una de las clases. La plataforma utilizada fue la misma anteriormente usada. La configuración de los *routers* no incluye la reserva explícita de ancho de banda (*bandwidth*). En su lugar es configurada la prioridad de la siguiente forma: en el caso B1, la clase 1 es configurada con prioridad

variable de 0 a 7 y la clase 2 es mantenida constante con prioridad 0 (*default*). En el caso B2, las dos clases varían de 0 a 7.

c) Equipos Utilizados

La plataforma de pruebas montada envuelve los siguientes equipos y productos: microcomputadoras, *routers*, instrumentos de prueba, cables etc, cada uno con las características detalladas a seguir.

- **Microcomputadoras**

Como cliente servidor fueron utilizados 04 microcomputadoras *desktop*, compatibles **IBM** (*International Business Machines*), Pentium 133 **MHz**, con *hard disk* suficiente para cargar el *Unix* y *Netperf*, y 64 MB de memoria **RAM** (*Random Access Memory*). Un *notebook Pentium Everex* para la configuración de los *routers*. El mínimo de *hard disk* utilizado es de 540 MB. La tarjeta de red utilizada en cada uno de ellos, es de 10 Mb/seg, 10 base T.

- ***Routers***

Fueron utilizados 3 *routers* marca *Cisco* modelo 3640 con 96 MB de memoria **RAM** y 32 MB de memoria *flash* con **IOS** (*Internetwork Operating System*) 3600 *software* (C3640-JS-MZ 122-11 T6 *bin*) *release software* (fc1) que soporta **MPLS**, **TE** y **QoS**. Cada *router* tiene 2 puertos seriales hasta 2 Mb (V35), un puerto **LAN** de 10 Mb 10 base T y consola para configuración.

- **Instrumentos de Medición**

El *software Netperf* (*Network Performance Benchmark*), versión 2.2 de la *Information Networks Division of Hewlett Packard* fue utilizado en todas las mediciones de *throughput* y como "carga" para simulación de congestión de los vínculos cuando necesario. Para las mediciones de retraso, junto con el comando *ping*, fue utilizado el *software Iperf*, ver. 1.6.4, desarrollado por el **NLANR**, y que puede ser bajado de la **URL** (*Uniform Resource Locator*): www.nlanr.net. Para la evaluación comparativa de algunas mediciones fue utilizado también el instrumento Domino *Internetwork Analyzer* modelo Domino **LAN**, de la *Wandel & Golterman*.

- **El *software Netperf***

Netperf es un "*benchmark*" que es utilizado para medir varios aspectos de *performance* de redes. Su principal objetivo es medir el rendimiento en función de la cantidad (*bulk*) de transferencia de datos y *request/response*, utilizando tanto paquetes de **TCP** como **UDP** y la *interface* de zoquetes de *Berkeley* **BSD** (*Berkeley Software Distribution*). Aunque originariamente de la *Hewlett-Packard*, este *software shareware* es mantenido e informalmente soportado por la **IND** (*Information Networks Division*) *Networking Performance Team* y no tiene ninguna responsabilidad de la **HP**. El usuario es libre para hacer mejorías y modificaciones en esta herramienta.

Netperf puede ser *downloaded* a través de **FTP** de *Internet*, del *site* [www.netperf.org/netperf].

- **Instalación del *Netperf:*** *Netperf* debe ser instalado sobre una plataforma *Unix* (en el caso de esta experiencia fue utilizado *OpenBSD*). Queda por cuenta del usuario escoger la versión más adecuada para su instalación particular. *Netperf* está proyectado en el modelo cliente-servidor, y por tanto existen dos programas ejecutables: *Netperf* y *Netserver*. Generalmente en las pruebas el lado cliente ejecuta el programa *Netperf* y el programa *Netserver* es llamado por el otro lado. Más detalles de este *software* pueden ser encontrados en el *site* del *Netperf* indicado anteriormente.

- **Componentes "*default*" del *Netperf:*** El *Netperf* está disponible en la *Internet* con los siguientes archivos:

Cuadro 3.14: Componentes del *Netperf*

Archivo	Descripción/Default Port: 12865
Netperf	Ejecutable
Netserver	Ejecutable
tcp-range-script	Socket Size = 32768 Bytes (fijo) Paquetes Tx = 1, 4, 16, 64...64 Kbytes
tcp-rr-script	Socket Size = 0 Bytes (utilizara default del sistema)
tcp-stream-script	Socket Sizes de: 57344, 32768 e 8192 Bytes
snapshot-script	Conjunto de pruebas TCP e UDP (instantáneo)
udp-rr-script	Socket Size = 0 (utilizara default del sistema)
udp-stream-script	Socket Sizes: 32768 Bytes Paquetes (Tx): 64, 1024, 1472 Bytes

***Scripts* Utilizados:** A continuación una breve descripción de los *scripts* utilizados en esta investigación.

TCP *Stream Script:* Este *script* es normalmente utilizado para medir "que tan rápido" un sistema puede enviar datos a otro y/o "que tan rápido" el otro sistema puede recibirlos (*throughput*). También es referenciado como "*stream*" *performance* o "unidireccional *stream*" *performance*. En esta investigación fue utilizado para las mediciones de tráfico transportado (*throughput*) y como "carga" para simulación de situaciones de congestión, modificado para las características siguientes:

Simulación de carga "tcp_60" modificado para:

Send/Receive Socket Size = "8192 8192 8192 8192 8192 8192 8192" *bytes.*

Send Size = "1448" *bytes.*

Mediciones de "tráfico transportado" "tcp_1448" modificado para:

Send/Receive Socket Size = "64, 128, 256, 512, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192" *bytes.*
Send Size = 1448 *bytes.*

d) Mediciones realizadas.

En cada experiencia fueron medidos los parámetros: tráfico transportado y retraso.

- **Mediciones de tráfico transportado**

Estas tienen por objetivo verificar el *throughput* en cada una de las experiencias efectuadas. Se utilizo el *Netperf* los scripts con la topología mencionada en la sección correspondiente a cada experiencia.

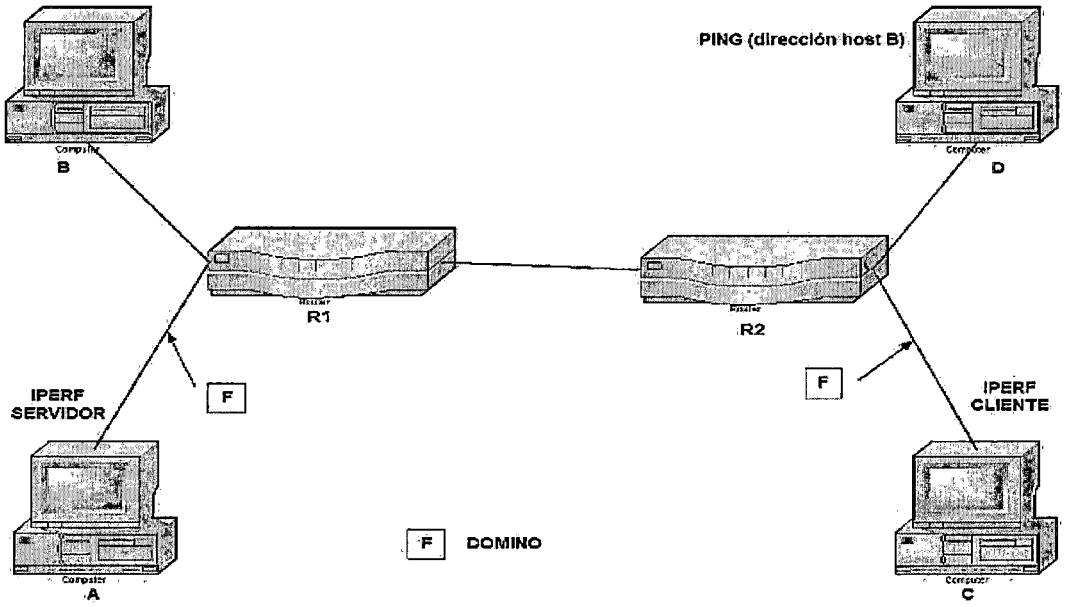
- **Mediciones de Retraso**

Todas las mediciones de **RTT** fueron efectuadas utilizando el *Iperf* como generador de tráfico y el comando *ping*. La "carga" de un flujo fue simulada con el comando del *Iperf*: *Iperf* -c (*host*) -u -i 1 -b (x) (dónde x = ancho de banda de carga, *host* = dirección **IP** del servidor). El servidor siempre en la situación: *Iperf* -s -u -i 1. Para las medidas fue utilizado el comando *ping* del *Windows*.

- **Topología para pruebas de retraso**

La Fig. 3.28 abajo muestra los equipos utilizados para las mediciones de retraso.

Fig. 3.28: Topología para pruebas RTT



3.4 Operacionalización de variables

Variables independientes= IPDT, IPDV,IPLR

QoS= f(IPDT,IPV,IPLR)

QoS= Calidad de Servicio.

3.5 Población y muestra

No se aplica.

3.6 Técnicas e instrumentos de recolección de datos

Las técnicas fueron lãs standares para medicion de lãs métricas de QoS y los instrumentos, basicamente softwares a ser configurados em cada equipo

3.7 Procedimientos de recolección de datos

Los procedimientos fueron los resultados directos de la configuración efectuada en cada caso, medidos con las herramientas físicas o software aplicativo.

3.8 Procesamiento estadístico y análisis de datos

Los datos fueron recogidos en tablas y posteriormente graficados como puede observarse en el trabajo.

CAPITULO IV

RESULTADOS

En este Capítulo son presentados y graficados los resultados de las experiencias citadas anteriormente. Los resultados de las mediciones de tráfico transportado (*throughput*) y retraso (*delay*) son comparados con los resultados teóricamente esperados y en el Capítulo siguiente son analizados y discutidos cada uno de los resultados obtenidos.

4.1 Experiencia 1: IP convencional (uno y dos flujos simultáneos)

4.1.1 Topología

Para cada experiencia, de acuerdo a las Fig. 3.22 a 3.27.

4.1.2 Configuración de los *Routers*

Todo el tráfico es encaminado entre los *routers* 1 y 2. Los detalles de configuración están indicados en cada experiencia.

4.1.3 Procedimiento de la prueba

Un o varios flujos en el *enlace* de X Mbps.

- **Tráfico Transportado:** La computadora A transmite para la computadora C el *script* tcp_1448 del *Netperf*, mientras las computadoras B y D permanecen inactivas. **RTT**: computadora A para computadora C: *Iperf* –u –i 1 –t60 –b (variable). Computadora B para computadora D: *ping* (dirección **IP** del *host*) –n 30.

4.1.4 Dos flujos simultáneos en el *enlace* de 2 Mbps

- **Tráfico Transportado:** La computadora A transmite para la computadora C y la computadora B para la computadora D, simultáneamente utilizando tcp_1448. **RTT**: la computadora A para la computadora C: *Iperf* –u –i 1 t60 –b (variable). La computadora B para la computadora D: *ping* (dirección **IP** del *host*) –n 30.

4.1.5 Gráficos resultantes

Se muestra a continuación los gráficos resultantes de la experiencia realizada.

Fig. 4.1: Un flujo IP convencional

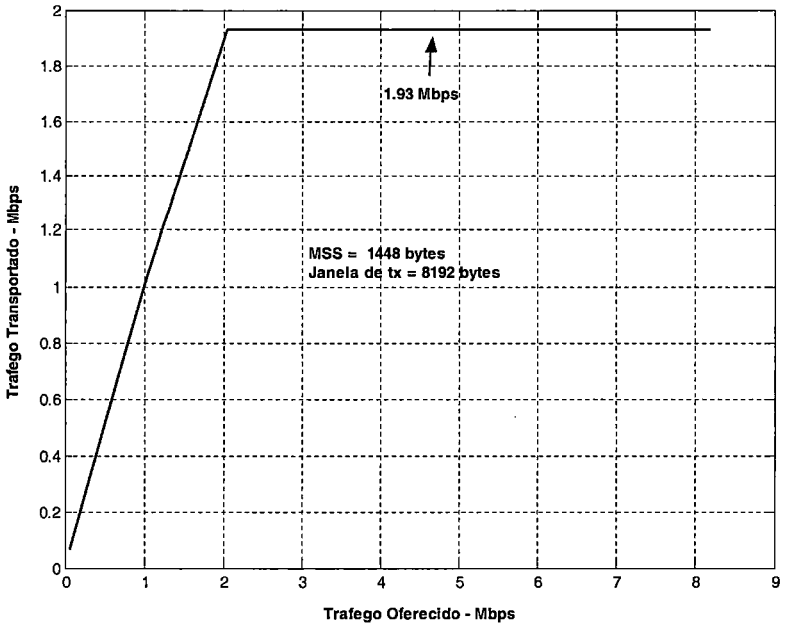


Fig. 4.2: Dos flujos IP convencionales

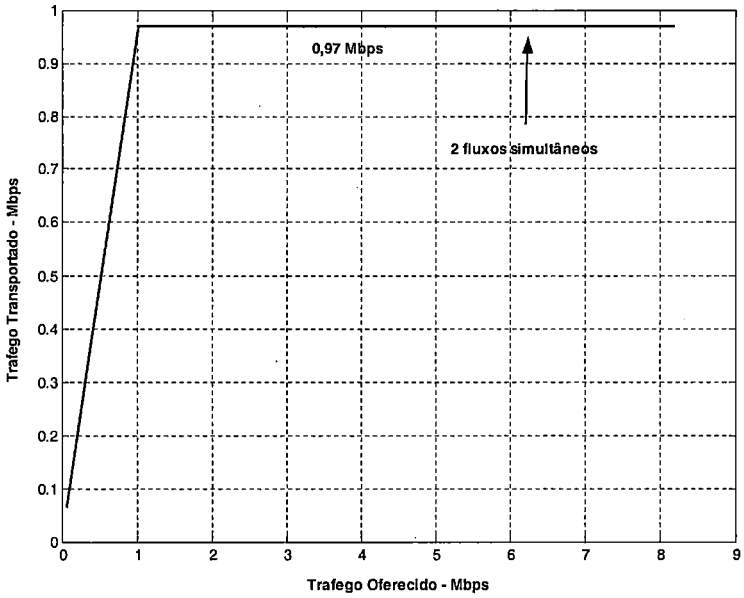
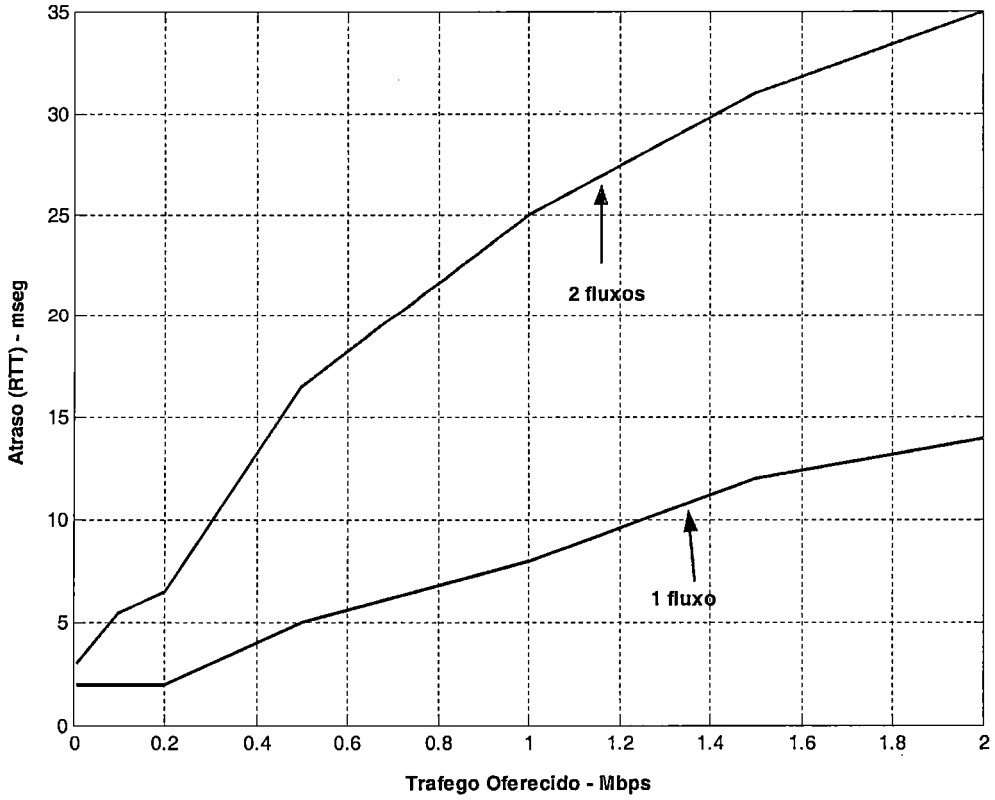


Fig. 4.3: RTT del IP convencional



Los gráficos de las Fig. 4.1 y 4.2 muestran que el tráfico transportado aumenta en la misma proporción del tráfico ofrecido, hasta un cierto límite impuesto por el límite de la capacidad de la *interface* utilizada, saturando prácticamente después de este límite. En el caso de la Fig. 4.1 en 1.93 **Mbps** (*throughput*) y en el caso de la Fig. 4.2 en 0.97 **Mbps**. El gráfico de la Fig. 4.3 muestra la variación del **RTT** para un flujo en la banda total de 1.93 **Mbps** y para dos flujos con media banda para cada (0.97 **Mbps**).

4.2 Experiencia 2: MPLS básico (uno y dos flujos)

4.2.1 Topología

La topología es idéntica a la experiencia anterior.

4.2.2 Configuración de los *Routers*

En este caso la configuración de los routers permite que todo el tráfico de datos sea encaminado por los routers 1 y 2.

4.2.3 Procedimiento de la Prueba

- Parte A: De acuerdo a lo indicado en la experiencia.
- Parte B: De acuerdo a lo indicado en la experiencia.

4.2.4 Gráficos resultantes.

Se muestra a continuación los gráficos resultantes de la experiencia realizada.

Fig. 4.4: 1 y 2 flujos MPLS

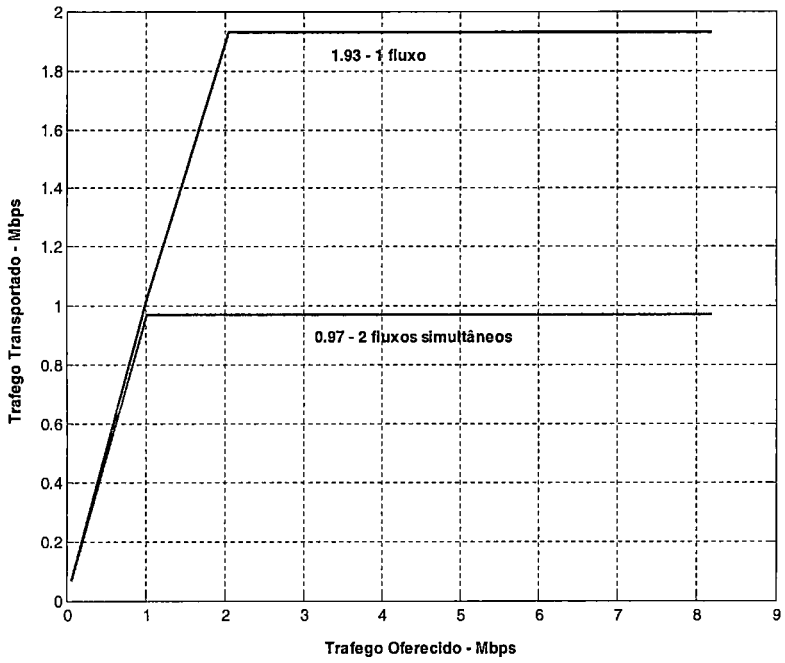
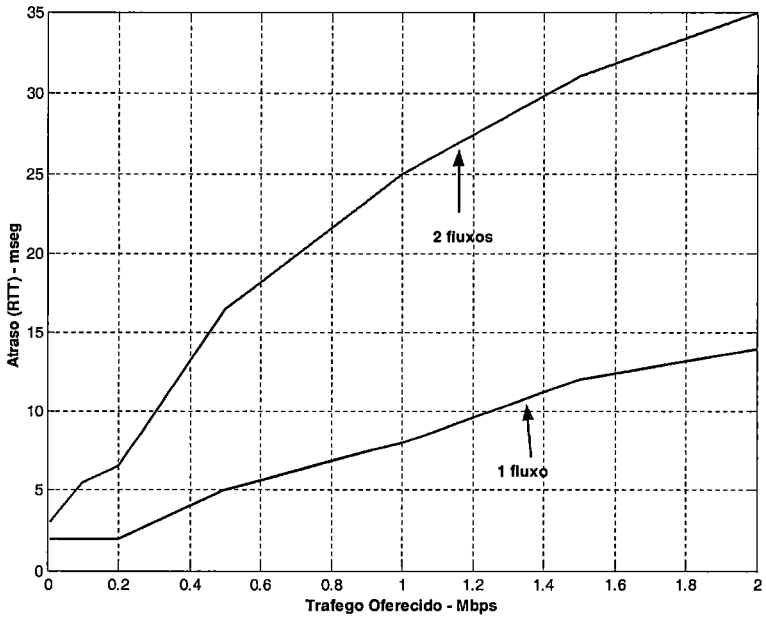


Fig. 4.5: RTT MPLS



4.3 Experiencia 3A: Caminos explícitos (uno y dos flujos)

4.3.1 Topología

Idéntico a la experiencia anterior.

4.3.2 Configuración de los *Routers*

En esta experiencia el tráfico ha sido encaminado entre los *routers* 1 y 2 a través del **LSP**1 y **LSP**2.

4.3.3 Procedimiento de la prueba parte A

Dos caminos explícitos, un flujo de datos.

- Tráfico Transportado: El computador A transmite para el C utilizando el script tcp_1448. **RTT**.

4.3.4 Procedimiento de la prueba parte B

Para esta prueba se ha considerado dos caminos explícitos y dos flujos de datos.

- Tráfico Transportado: El computador A transmite para el C y B transmite para D simultáneamente utilizando el script tcp_1448 para cada uno de ellos.

4.3.5 Gráficos resultantes.

Se muestra a continuación los gráficos resultantes de la experiencia realizada.

Fig. 4.6: Caminos explícitos, 1 y 2 flujos

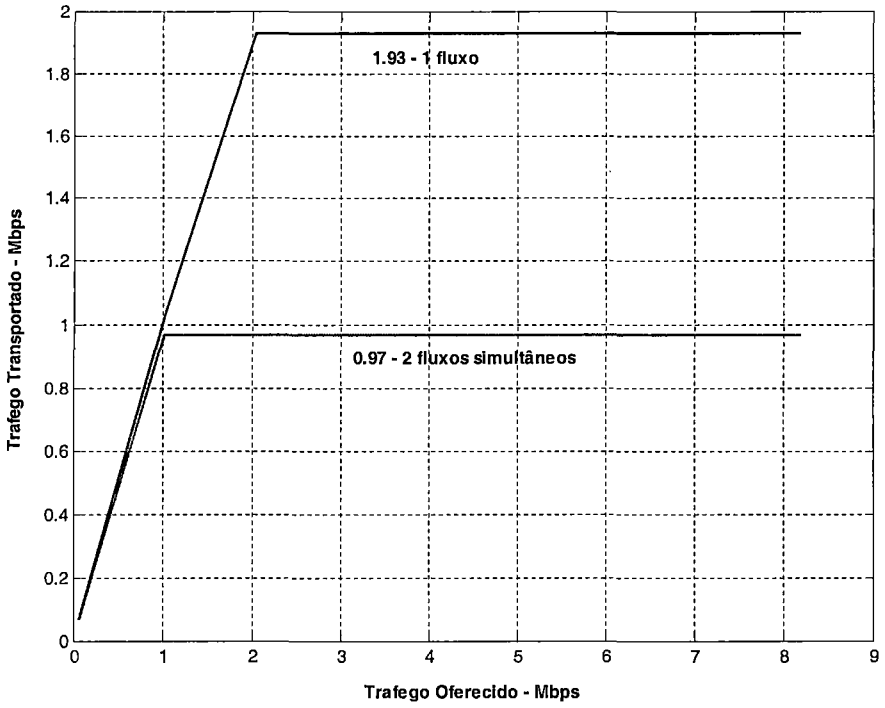
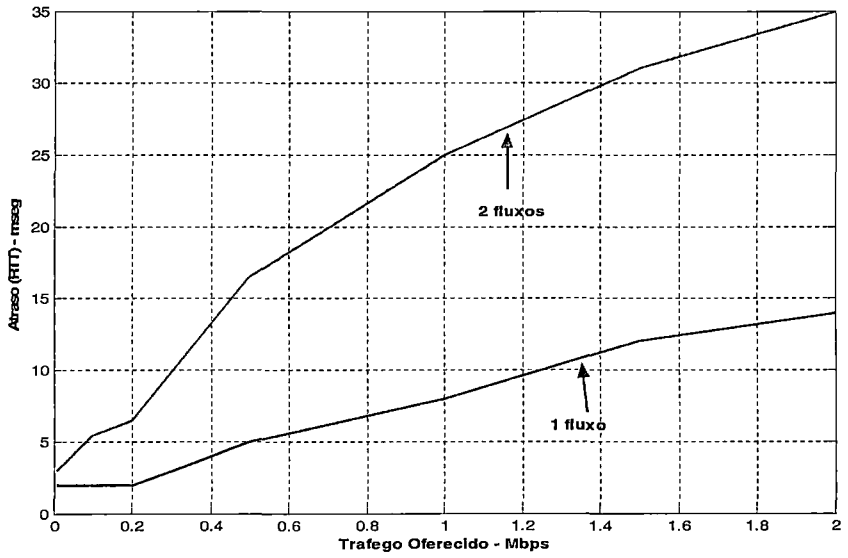


Fig. 4.7: RTT, caminos explícitos



4.4 Experiencia 3B: Dos caminos explícitos y dos flujos diversificados

4.4.1 Topología

Es la misma utilizada en la experiencia anterior.

4.4.2 Configuración de los *Routers*

El tráfico de la computadora A es encaminado a través del **LSP**1 y el tráfico de la computadora B es desviado para el nuevo **LSP**2.

4.4.3 Procedimiento de la prueba

Es igual a la anterior experiencia.

4.4.4 Gráficos resultantes.

Se muestra a continuación los gráficos resultantes de la experiencia realizada.

Fig. 4.8: Caminos explícitos y dos flujos diversificados

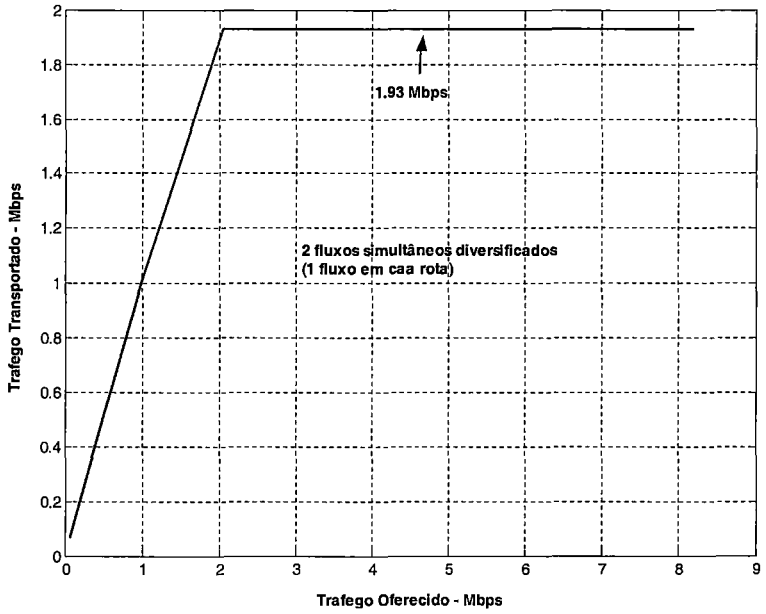
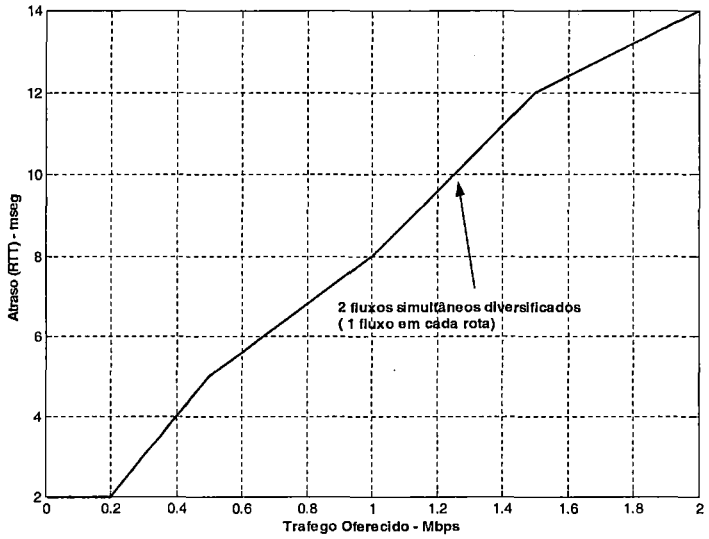


Fig. 4.9: RTT de dos flujos diversificados en caminos explícitos



El gráfico de la Fig. 4.8 muestra que aún teniendo 2 flujos simultáneos, aun encaminados por rutas diferentes, usan la totalidad de la banda (1.93**Mbps**), en cada ruta.

La variación del **RTT** es mostrada en la Fig. 4.9.

4.5 Experiencia 4A: CoS y reserva de ancho de banda

4.5.1 Topología

La topología empleada es igual a las utilizadas en las experiencias anteriores.

4.5.2 Configuración de los *Routers*

Todo el tráfico es encaminado entre los *routers* **LSR**1, **LSR**2 y **LSR**3.

4.5.3 Procedimiento de la prueba

- Clase 1: Paquetes entrando por la *Ethernet* 0/0 y originados en el *host* A.
- Clase 2: Paquetes entrando por la *Ethernet* 0/1 y originados en el *host* B.
- Tráfico Transportado: A transmite para C y B transmite para D simultáneamente utilizando *script* "tcp 60". **RTT**:

4.5.4 Graficos resultantes.

Se muestra a continuación los gráficos resultantes de la experiencia realizada.

Fig. 4.10: Reserva de Banda

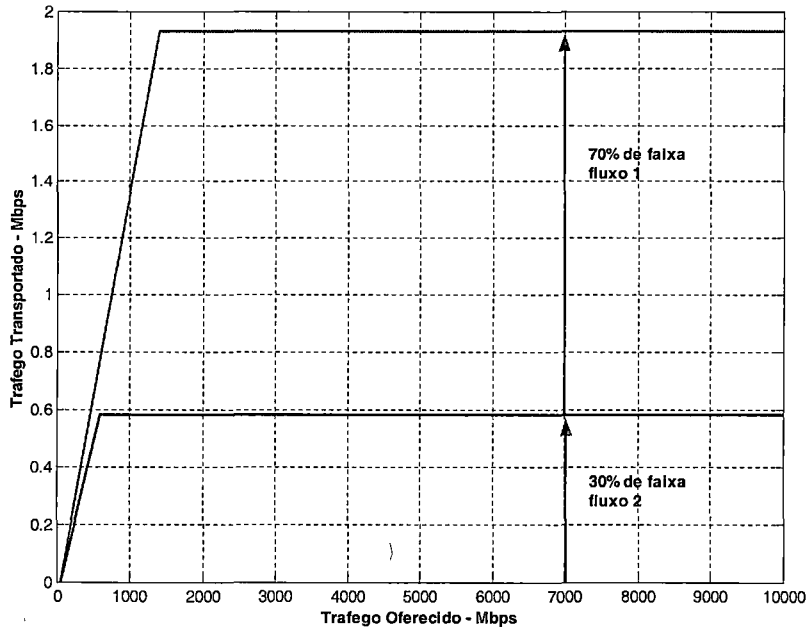
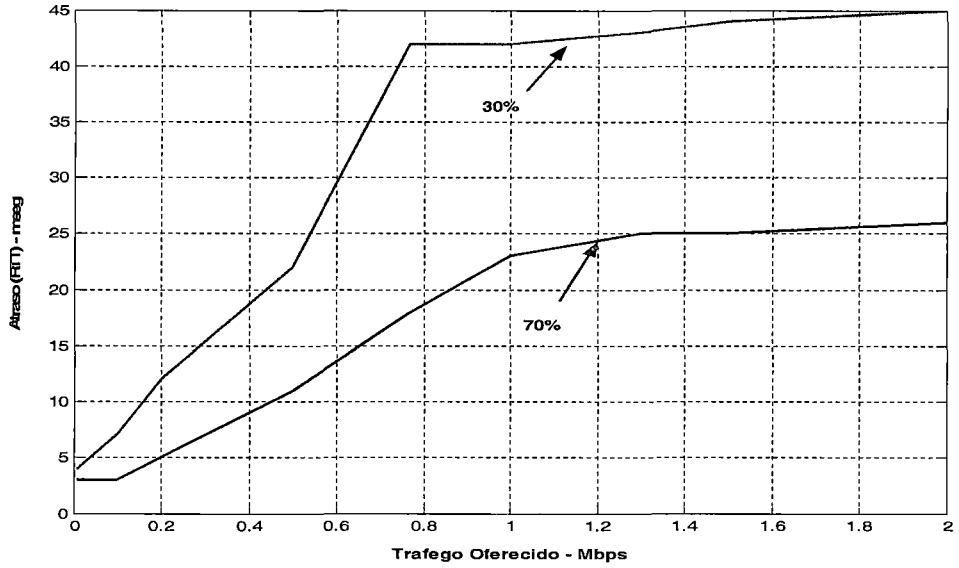


Fig. 4.11: RTT reserva de banda



El gráfico de la Fig. 4.10 muestra claramente las bandas asignadas para los 2 flujos con 30% y el otro con 70% de la banda total. La variación del **RTT** para las mismas bandas está indicada en el gráfico de la Fig. 4.11.

4.6 Experiencia 4B: CoS y prioridad IP

4.6.1 Topología

La topología es idéntica a las usadas en las experiencias anteriores

4.6.2 Configuración de los *Routers*

En esta experiencia se ha considerado que todo el tráfico es encaminado entre los *routers* **LSR**1, **LSR**2 y **LSR**3.

4.6.3 Procedimiento de la prueba B1

Para esta prueba se considero la clase 2 sin prioridad y clase 1 con prioridad variable.

- **Tráfico Transportado:** A transmite para C y B para D simultáneamente utilizando el *script* tcp 60. **RTT.**

4.6.4 Procedimiento de la prueba B2

En esta prueba se considero la clase 1 y clase 2 con prioridad variable.

- **Tráfico Transportado:** El computador A transmite para C y B para D simultáneamente, utilizando el *script* tcp 60. La clase 1 y clase 2 configuradas con prioridad variable de 0 a 7. El **RTT** es igual a la experiencia anterior.

4.6.5 Gráficos resultantes

Se muestra a continuación los gráficos resultantes de la experiencia realizada.

Cuadro 4.1: Resultados de las mediciones de prioridad IP

Prioridad								
Clase	0	1	2	3	4	5	6	7
Clase 1Mbps	0,97	1,29	1,45	1,55	1,61	1,65	1,69	1,72
% *	50	66,8	75,1	80,3	83,4	85,5	87,5	89,1
Clase 2Mbps	0,97	0,65	0,49	0,39	0,32	0,28	0,24	0,22
%	50	33,2	24,9	19,7	16,6	14,5	12,5	10,9

* Porcentaje indica el valor en **Mbps** en relación al tráfico máximo transportado (**Mbps**/1,93) = %.

Fig. 4.12: Prioridad 0

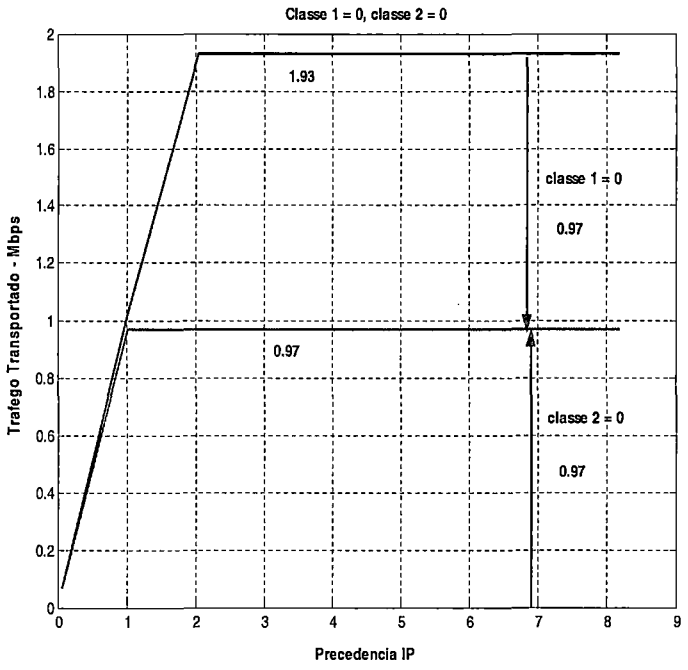


Fig. 4.13: Prioridad 1

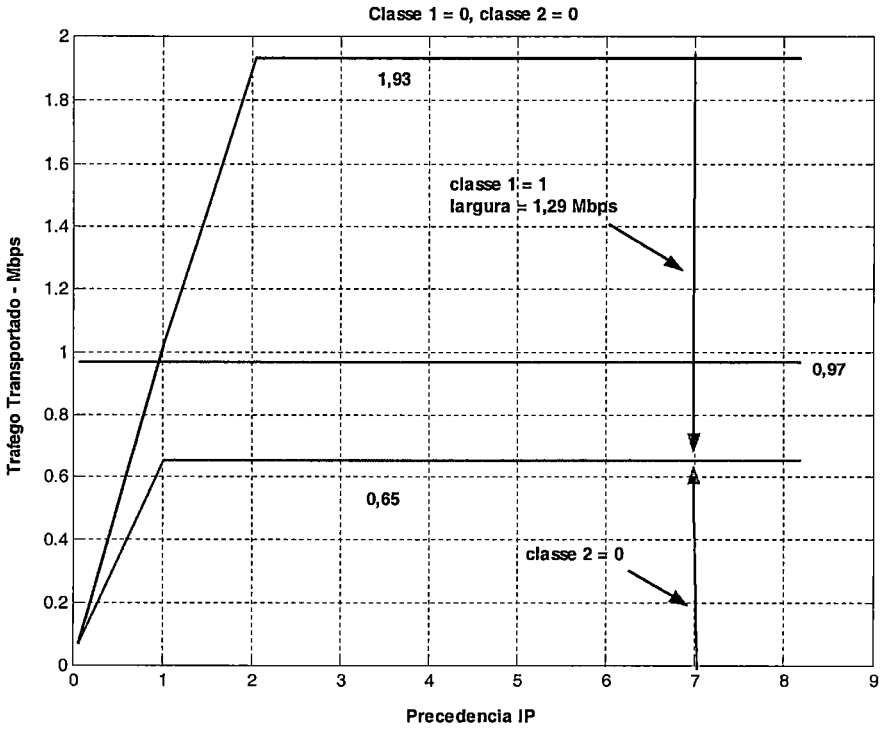


Fig. 4.14: Prioridad 2

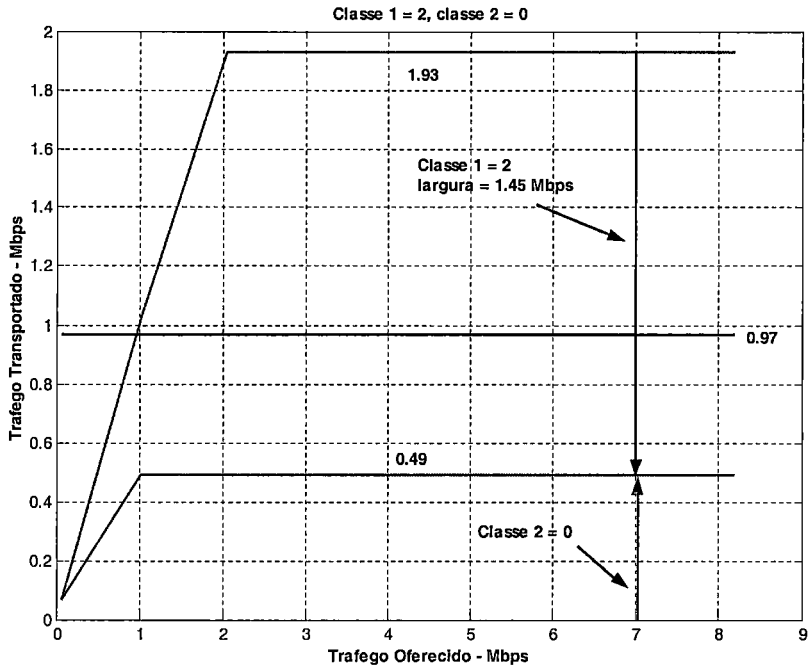


Fig. 4.15: Prioridad 3

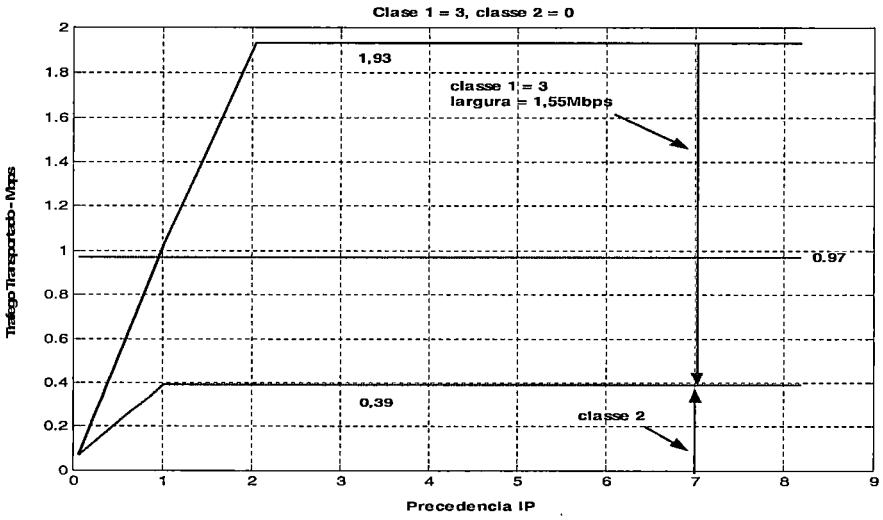


Fig. 4.16: Prioridad 4

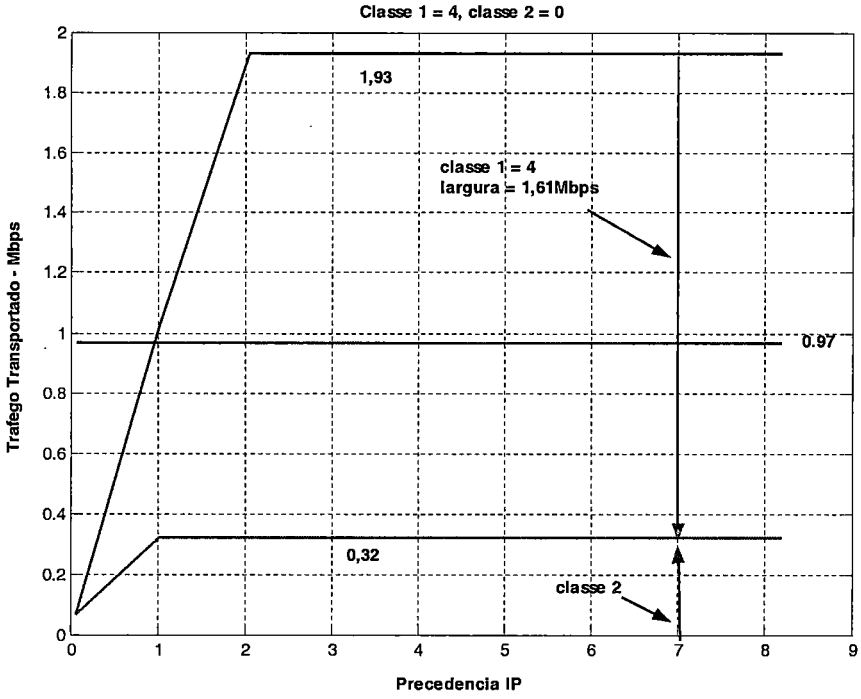


Fig. 4.17: Prioridad 5

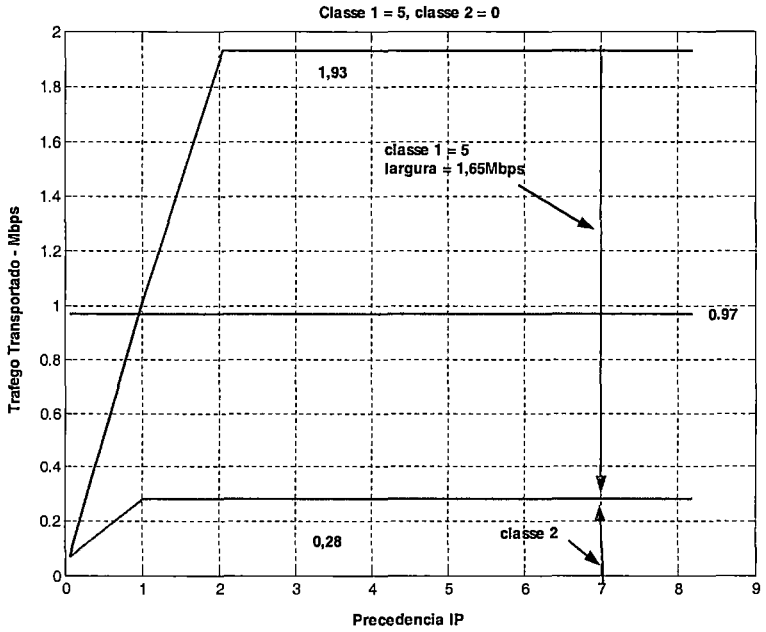


Fig. 4.18: Prioridad 6

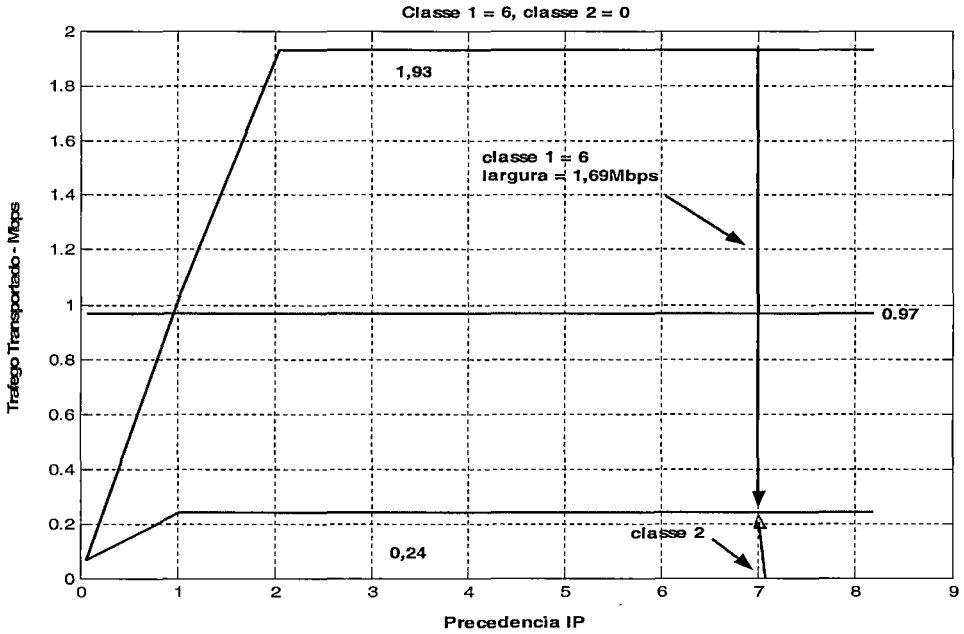
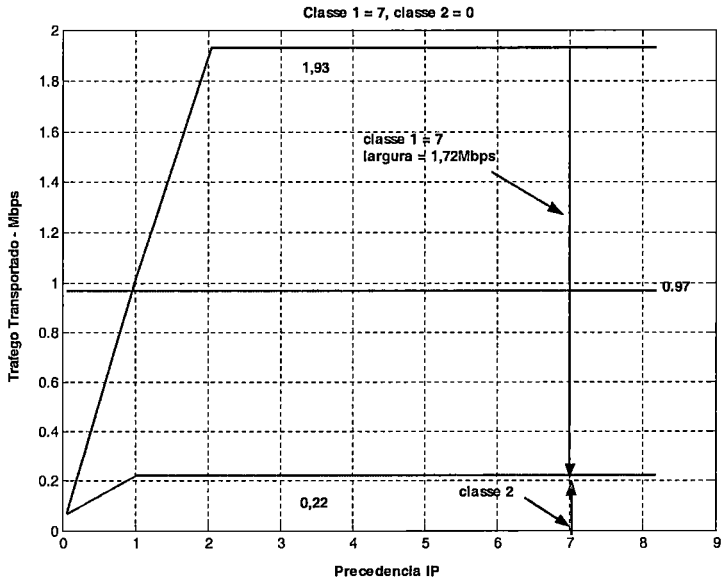


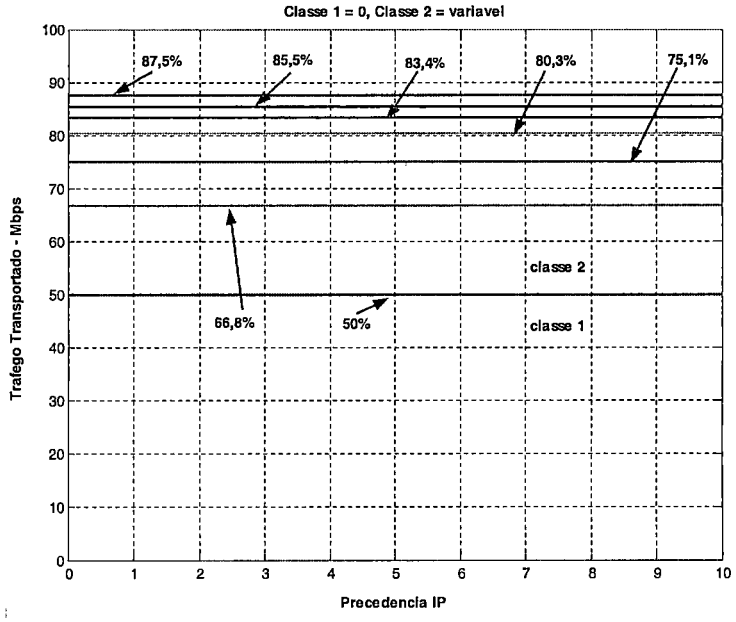
Fig. 4.19: Prioridad 7

Fig. 4.19: Prioridad 7



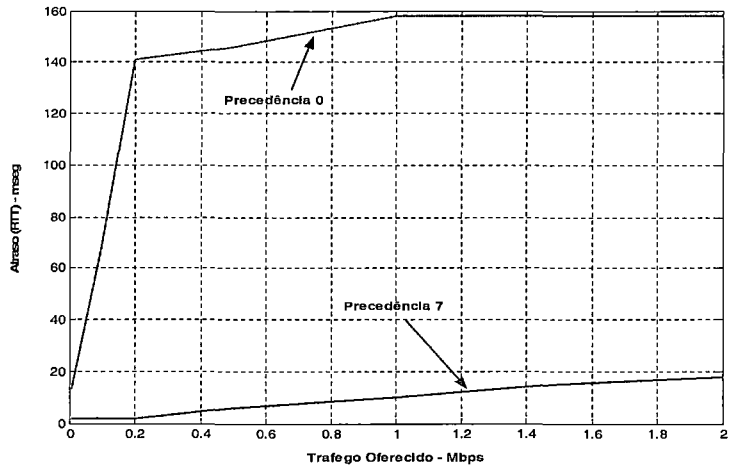
Los gráficos de las Fig. 4.12 a 4.19 muestran los efectos de la configuración de la prioridad **IP** en dos flujos simultáneos (clases) en un mismo enlace. La clase 2 sigue con prioridad fija igual a 0 y la clase 1 varia de 0 a 7. La Fig. 4.13 muestra el efecto de la configuración de la clase 1 con prioridad 1. Vea el crecimiento de la banda de la clase 1 a 1.29 **Mbps** desde 0.97 **Mbps,** y la reducción de la clase 2 a 0.65 **Mbps**.

Fig. 4.20: Prioridades 0 a 7 (1 flujo fijo)



El gráfico de la Fig. 4.20, es el resumen en porcentaje de los valores de incremento de banda, cuando se establece la prioridad **IP** para un flujo de 0 a 7 y el otro permanece sin prioridad.

Fig. 4.21: RTT prioridades 0 a 7



El gráfico de la Fig. 4.21 muestra la variación de **RTT** para las bandas resultantes de la prioridad 0 y 7 en la clase.

- **Gráficos resultantes**

Se muestra a continuación los cuadros y los gráficos resultantes de la experiencia realizada.

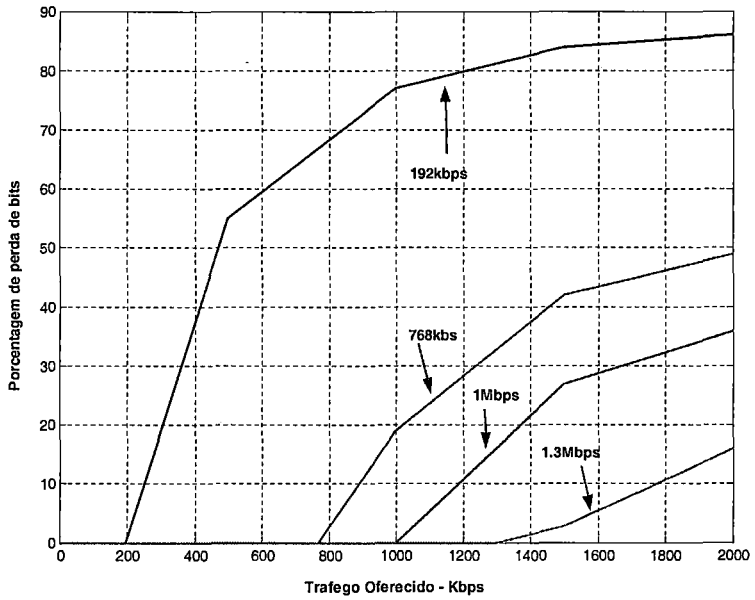
Cuadro 4.2: Prioridades en Mbps

Clase \ Clase	1	2	3	4	5	6	7
1	0,97	1,16	1,29	1,38	1,45	1,50	1,55
	0,97	0,78	0,65	0,55	0,49	0,43	0,39
2	0,78	0,97	1,11	1,21	1,29	1,35	1,41
	1,16	0,97	0,83	0,73	0,68	0,58	0,53
3	0,65	0,83	0,97	1,07	1,16	1,23	1,29
	1,29	1,11	0,97	0,86	0,78	0,71	0,65
4	0,55	0,73	0,86	0,97	1,06	1,13	1,19
	1,38	1,21	1,07	0,97	0,88	0,81	0,75
5	0,49	0,65	0,78	0,88	0,97	1,04	1,11
	1,45	1,29	1,16	106	0,97	0,89	0,83
6	0,43	0,58	0,71	0,81	0,89	0,97	1,03
	1,50	1,35	1,23	1,13	1,04	0,97	0,90
7	0,39	0,53	0,65	0,75	0,83	0,90	0,97
	1,55	1,41	1,29	1,19	1,11	1,03	0,97

Cuadro 4.3: Prioridades en porcentaje de la banda

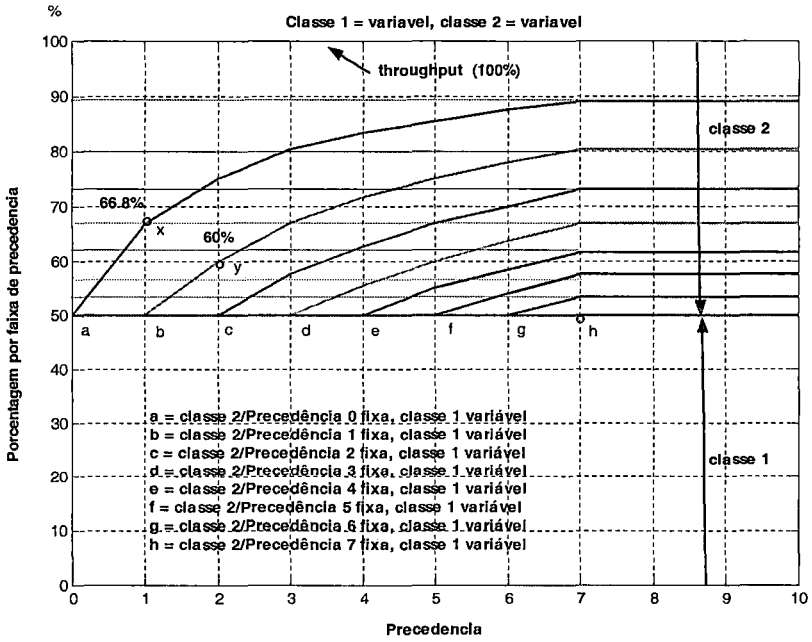
Clase	Clase	1	2	3	4	5	6	7
1		50						
2		60	50					
3		67	57,5	50				
4		71,5	62,7	55,4	50			
5		75,1	67	60	55	50		
6		78	70	63,7	58,5	53,9	50	
7		80,3	73	67	61,7	57,5	53,4	50

Fig. 4.22: Porcentaje de pérdida de *bits* en las mediciones RTT



El gráfico de la Fig. 4.22 muestra el resultado, en porcentaje, de la pérdida de *bits* en función del tráfico ofrecido, para algunas bandas asignadas, dentro de la capacidad de tráfico máximo permitido por la *interface* física utilizada (1.93 **Mbps**). Vea que solamente existe pérdida fuera de los límites máximos de las bandas, es decir, en la región de saturación.

Fig. 4.23: Prioridades 0 a 7 (Dos flujos variables)



El gráfico de la Fig. 4.23 muestra el resultado de la configuración de la prioridad **IP,** para 2 clases de flujos en un mismo enlace y ambas con prioridad variable. Este gráfico permite calcular con bastante aproximación, la asignación de banda para dos flujos dependiendo de la prioridad. Como ejemplo, el punto "x" es la intersección de la curva "a" de prioridad 0 con la recta vertical de prioridad 1. Este punto indica que "en situaciones de congestión", la clase configurada con prioridad 1 tomará para sí 66,8% del total de la banda y el restante quedará para la otra clase. Similarmente el punto "y", indica la asignación de 60% para una clase con prioridad 2 y 40 % para la otra clase con prioridad 1.

CAPITULO V

DISCUSION DE RESULTADOS

Todos los conceptos y protocolos presentados en esta investigación se refieren específicamente a la parte de la red **IP** conocida como "*backbone network*", para diferenciar de las técnicas y protocolos utilizados en la parte conocida como "*access network*" (red de acceso), que es aquella en que los usuarios conectan directamente sus servidores/clientes (computadoras/equipos de comunicación). Esto es, de los modelos de **QoS**, protocolos y mecanismos de control que actúan en la espina dorsal (núcleo) de las redes **IP**. Dos modelos definidos por el **IETF** fueron estudiados en esta investigación **IntServ** y **DiffServ.** El **IntServ** fue dejado un poco de lado debido a sus dificultades de implementación práctica, aunque algunos mecanismos fueron introducidos con esta arquitectura, como el **RSVP**, al haber tenido un gran desarrollo y participar actualmente de forma intensiva en la implementación del modelo **DiffServ**. No está totalmente desechado y la expectativa es que nuevos mecanismos en desarrollo lo traigan nuevamente al escenario de estas implementaciones. Actualmente, hablar de **QoS** significa básicamente referirse a la arquitectura **DiffServ**.

Por la línea general de desarrollo de esta investigación, este Capítulo final tendrá sus comentarios divididos en dos partes, una acerca de la teoría de implementación de **QoS** con **MPLS** y la otra acerca de la parte experimental efectuada.

5.1 Implementación de QoS con MPLS

En esta parte fueron destacados principalmente, dos asuntos: primero, las características básicas de la implementación del **DiffServ** a través de **MPLS** y, segundo, las ventajas adicionales de la utilización del mismo.

En cuanto a las características de implementación, los puntos más relevantes están relacionados con las diferencias básicas entre una implementación directa **IP** y una implementación a través de **MPLS**:

- Con **MPLS** en el **LSR** de entrada, es insertado un encabezado **MPLS** en el paquete. El **DSCP** en el encabezado **IP** es asignado al **EXP** *field* del encabezado **MPLS**.
- Con **MPLS**, la clasificación **BA** es basada en el campo **EXP** en el lugar del **DSCP**.
- Con **MPLS** en el **LSR** de salida, el encabezado **MPLS** es eliminado.

El primero punto significa que con **MPLS**, los paquetes siguen teniendo su **DSCP** fijado en la entrada de la red, con la diferencia de que cuando son insertados los encabezados **MPLS**; los **DSCPs** en los encabezados **IP** son asignados al campo **EXP** del **MPLS** (*default*). Debido a que el **DSCP** es de longitud 6 *bits* y el campo **EXP** es sólo 3 *bits*, alguna información en el **DSCP** pueden ser perdídas en el asignamiento. Más, como ya fue citado en el Capítulo 3, sólo los 3 *bits* más a la izquierda del **DSCP** tienen información útil (*bits* de prioridad) y esos 3 *bits* pueden ser copiados al campo **EXP**. Los 2 más a la izquierda indican la clase de servicio y el tercer *bit* indica cuando el paquete está dentro del perfil o fuera de él. Esta forma de operación o asignamiento también es conocida como **E-LSP**.

Con relación al segundo punto, clasificación **BA**, podremos complementar que, en el medio de un **LSP**, esta clasificación es basada en el campo **EXP** en lugar del **DSCP**. Los gerenciamientos de *buffer* y *scheduling* de colas son idénticos, con o sin **MPLS**, y por consiguiente, el comportamiento de los paquetes por *link* (**PHB**) también es idéntico. Es decir, positivo, pues implica que los **ISPs** que proporcionan **QoS** con **MPLS** pueden fácilmente interoperar con **ISPs**, que proporcionan **QoS** sin **MPLS**, significando que la utilización de **MPLS** o no, en el ofrecimiento de **QoS** es transparente para los usuarios finales.

Entre las ventajas adicionales de la implementación con **MPLS**, podremos mencionar:

- **Múltiples LSPs**

Algunas veces es necesario permitir que diferentes clases de tráfico sigan caminos diferentes. Para hacer esto, pueden existir múltiples caminos del mismo

- **Múltiples LSPs**

Algunas veces es necesario permitir que diferentes clases de tráfico sigan caminos diferentes. Para hacer esto, pueden existir múltiples caminos del mismo origen al mismo destino, un **LSP** por clase. Adicionalmente, **LSPs** de clases diferentes pueden tener diferentes "restricciones". Cada *link* físico puede ser dividido en múltiples *links* virtuales para diferentes clases de tráfico. Por consiguiente, la red física puede ser dividida en múltiples redes virtuales, una por clase, con diferentes topologías y recursos, si es necesario. Al **LSP** *premium* se le puede dar alta prioridad, al **LSP** "*assured*", media prioridad y al **LSP** "*best effort*", baja prioridad. También el **LSP** *premium* podrá seguir los caminos más cortos, principalmente en *links* de alta velocidad dónde el retraso de propagación se torna una parte significante del retraso total y la utilización de caminos más cortos puede reducir significativamente ese retraso. En caso de falla de algún *link* o *router*, el tráfico *premium* podrá tener alta prioridad para utilizar recursos de "*backup*", en relación a las otras clases de tráfico. Este proceso puede ser automático o controlado por enrutamiento "*constraint-based*" (basado en restricciones). Aunque el enrutamiento basado en clases pueda también ser hecho en una red **IP** sin **MPLS**, el *router* de origen no puede controlar los caminos de su tráfico a cada destino, a menos que el camino completo sea indicado en el encabezado de cada paquete.

5.1.1 Asignación de Recursos

Utilizando una estructura basada en **MPLS** para proporcionar **DiffServ**, se torna mucho más fácil la asignación de recursos en un dominio de **ISP**:

a) Fijando el ancho de banda máxima reservable para tráfico de alta prioridad a un cierto nivel, y utilizando el enrutamiento "basado en restricciones" para prevenir la concentración de tráfico de alta prioridad, resolviendo así, un problema que la red **IP** - **DiffServ** no puede resolver por sí sola.

b) Las tasas de transmisión de todos los **LSPs** pasando a través de una *interface* proporcionan las estadísticas necesarias para fijar las nuevas tasas de colas. Con estos valores y con los valores de retraso de colas deseables, los parámetros de **RED/RIO** y el tamaño de las colas pueden fácilmente ser derivadas.

Con **MPLS**, cuando el número de flujos en tránsito se incrementa, el número de flujos en cada **LSP** también se incrementa, más el número de **LSPs** no necesariamente, proporcionando, así, "escalabilidad" en las grandes redes de **ISPs**.

5.1.3 Congestión

El servicio **DiffServ** sólo no puede proporcionar una calidad de servicio **QoS** suficientemente confiable. Proporciona una degradación diferenciada a tráficos diferentes "sólo" en las situaciones de congestión de red. Cuando no hay congestión, el desarrollo de todos los tráficos se mantendrá igual, como si no se utilizara **DiffServ** (no hace diferencia cuando no existe congestión en la red).

5.1.4 Continuidad de Servicio

Con **DiffServ**, cuando ocurre falla o queda de un *link* o *router fuera*, el tráfico en curso en ese instante se pierde, sin que se pueda hacer nada si ningún mecanismo de recuperación fue previsto. Con **MPLS/TE**, esto puede ser resuelto mediante el recurso de *fast reroute* de la Ingeniaría de Tráfico del **MPLS**.

5.2 Parte Experimental

Aquí fue tratado especialmente el asunto relacionado con Ingeniaría de Tráfico y **QoS** a través de la facilidad **CoS** disponible en el *software Cisco* utilizado. Considerando también que el **MPLS** permite la creación de *links* dedicados, incluso reserva de ancho de banda, además de la evaluación de desarrollo utilizando recursos de **TE**, fueron evaluados también los efectos de diferenciación de servicio de **QoS** configurando prioridad **IP**. Fueron utilizadas las *interfaces* de entrada *Ethernet* 0 y 1 como punto de selección del tráfico y clasificación. Los resultados obtenidos refuerzan los fuertes argumentos sobre la efectividad de los recursos del **MPLS** y muestran claramente que los recursos de **TE** y diferenciación de servicio mejoran el desarrollo y escalabilidad de las redes *backbone* **IP**. Estos aspectos de mejoría de desarrollo son mucho más significativos en las situaciones de congestión de la red.

A continuación se muestra una evaluación de los resultados de las experiencias efectuadas.

En las situaciones convencionales de redes **IP** (operando sin ningún mecanismo de reserva de banda), en un *link* de 2**Mbps** (o de otra magnitud), un único flujo en el *link* asigna para si toda la banda disponible. Si un segundo flujo aparece, la banda será dividida por los dos, si un tercero aparecer, y así sucesivamente. En el caso de estas experiencias, la *interface* **WAN** utilizada permite una transferencia máxima de 2**Mbps** nominal, pero, en el conjunto, permite un tráfico transportado máximo de 1.93**Mbps** (*throughput* del sistema de testes). Con dos flujos simultáneos en el *link*, el **TMT** (tráfico máximo transportado) para cada uno de los flujos baja para 1.93/2 = 0.965**Mbps**. Como el *Netperf* cambia este valor para 0.97**Mbps**, adoptamos los valores de 1.93 para la banda total y 0.97**Mbps** para media banda en todas las referencias de **TMT** para un y dos flujos respectivamente.

La experiencia 1 presenta como resultados las referencias de **TMT** y retraso (**RTT**) para las otras experiencias. El *enlace* está basado en los protocolos *frame relay* y **OSPF**. Se muestra el tráfico transportado y retraso para dos situaciones típicas, una cuando un único flujo cruza el *enlace* de 2 **Mbps** con *throughput* máximo de 1.93**Mbps** y el correspondiente retraso, y la otra situación cuando dos flujos simultáneos cruzan el mismo *enlace* (0.97**Mbps**).

La experiencia 2 muestra que sólo la habilitación del protocolo **MPLS** en si no tiene ningún efecto de mejora en relación a la situación convencional. Se espera con **MPLS** una conmutación de paquetes más rápida que las convencionales, pero esto no fue verificado en esta experiencia.

La experiencia 3 presenta como resultado, la eficacia de los mecanismos de Ingeniaría de Tráfico para evitar situaciones de congestionamiento, al permitir enrutamiento de flujos/clases por enlaces menos, o no, congestionados. No se observa ninguna mejora en relación a la situación convencional, cuando uno o dos flujos cruzan el mismo enlace, y por tanto, dividiendo la banda total del *enlace* entre los dos flujos (0.97**Mbps** para cada uno). Se muestra que, desviando el tráfico para otras rutas, en el caso para una ruta libre para el mismo destino, aunque con un enlace adicional, resulta beneficioso para los dos flujos.

tráfico para otras rutas, en el caso para una ruta libre para el mismo destino, aunque con un enlace adicional, resulta beneficioso para los dos flujos.

La experiencia 4A muestra, que el mecanismo de reserva de ancho de banda (en el caso 70% y 30% de la banda disponible) funciona efectivamente en situaciones de congestión, y por tanto, permite asegurar la banda necesaria. En la práctica, es posible reservar hasta 75% de la banda disponible, es decir, si la interface es 2**Mbps** nominal, se puede reservar hasta 1.5**Mbps** (2 x 0.75), recordando solamente que la banda real reservable es 1.93 x 0.75 = 1.4475**Mbps**.

La experiencia 4B se refiere a la clasificación de dos flujos de la misma forma que en la experiencia 4A, basados en la interface de entrada (**ETH** 0/0 clase 1 y **ETH** 0/1 clase 2), solamente con habilitación de la prioridad **IP** para ambas las clases, sin reserva explícita de recursos. Se muestran los efectos de la prioridad **IP** en la asignación de banda por flujo para el caso de dos flujos cruzando simultáneamente el enlace de interface nominal de 2**Mbps** y *throughput* máximo de 1.93**Mbps**. En la parte B1, la clase 2 está configurada sin prioridad, o, equivalentemente, tiene fijada la prioridad default igual a 0. Observamos que, partiendo de la situación inicial (medio ancho de banda para cada uno de los flujos o 0.97**Mbps** para cada), la banda de la clase 1 crece de 0.97 para 1.29**Mbps** (0.32**Mbps**) cuando es configurada con prioridad 1 y decrece el mismo valor para la clase 2. La clase 1 con prioridad 2 crece para 1.45**Mbps** (1.45 – 0.97 = 0.48**Mbps**) y la clase 2 decrece para 0.97 – 0.48 = 0.49**Mbps** y así sucesivamente (Fig. 4.12 a 4.19). La Fig. 4.20 presenta la misma condición antes mencionada, con sus valores en porcentaje del *throughput* máximo (1.93**Mbps**), que pueden ser de ayuda para extrapolar otros valores de *interface* diferentes de los aquí analizados (2**Mbps**). El cuadro 5.1 abajo muestra los valores obtenidos en las mediciones efectuadas versus los valores calculados.

Cuadro 5.1: Valores medidos y calculados

PRIORIDAD	1	2	3	4	5	6	7
MEDIDO	1.29	1.45	1.55	1.61	1.65	1.69	1.72
CALCULADO	0.666 x 1.93 = 1.285	0.750 x 1.93 = 1.447	0.80 x 1.93 = 1.544	0.833 x 1.93 = 1.607	0.857 x 1.93 = 1.654	0.875 x 1.93 = 1.688	0.888 x 1.93 = 1.713

En la Fig. 4.23 se mantiene fija una clase y se varía la prioridad de la otra, de 1 a 7. Las letras a, b, c, d, e, f, g, h indican el inicio de las curvas de prioridad (a = 0 hasta h = 7). En el eje de las abscisas, están indicadas las prioridades. Con este gráfico, podremos calcular el aumento o disminución de banda cuando se configura la prioridad para dos flujos. Por ejemplo, el punto "x" (intersección de la curva de prioridad 0 con la prioridad 1) indica que el flujo con prioridad 0, aún compitiendo con el flujo de prioridad 1, en el mismo *link* tendrá su banda disminuida (y el otro aumentada) en 16,8% durante las situaciones de congestión. Similarmente, un flujo con prioridad 1 (punto "y"), aún compitiendo con un flujo de prioridad 2, tendrá su banda disminuida en 10%. Observamos, también, que la mayor "expansión" de banda se consigue para una clase con prioridad 0 y la otra con prioridad 7 (aproximadamente 40%).

La Fig. 4.22 muestra de forma esperada de la pérdida de bits en función del tráfico ofrecido. Observamos que, para los anchos de banda indicadas 192**Kbps**, 768**Kbps**, 1**Mbps** y 1.3**Mbps**, la pérdida de *bits* comienza sólo después los límites finales de las bandas. No existe pérdida de *bits* interna en la banda. Similarmente, el comando *ping* no ha sufrido pérdida ninguna en toda la banda hasta 2**Mbps**. Esta medida fue efectuada utilizando el *Iperf* – **UDP** y el comando *ping*.

CONCLUSIONES

Como un corolario de esta investigación se puede concluir lo siguiente:

a).-Las investigaciones en curso están tentando reactivar y mejorar la escalabilidad del **RSVP** de modo de permitir que la arquitectura **IntServ** pueda agregar mayores reservas de recursos y, potencialmente, participar de modo más significativo en las grandes redes **IP**.

b).-Actualmente, la arquitectura **DiffServ** es la alternativa que más proporciona la escalabilidad necesaria cuando se pretende ofrecer en las grandes redes **IP**, servicios mejores que el "*best effort*".

c).-El **MPLS** tiene soporte total para las dos arquitecturas arriba mencionadas.

d).-La Ingeniería de Tráfico del **MPLS** reduce la congestión y optimiza el uso de los recursos de las redes existentes, permitiendo el gerenciamiento cuidadoso del tráfico a través de la red.

e).-Todos los ítems antes mencionados imponen al **MPLS** una "llave" en el desarrollo e implementación de nuevos servicios, dirigidos principalmente a aplicaciones en tiempo real.

RECOMENDACIONES

a).- La comunicación en las grandes redes IP por lo general se encuentran saturada por la gestión de múltiples servicios a los usurarios. Por lo tanto la propuesta se enmarca en mejorar la comunicación utilizando el protocolo RSVP.

b).-Una mejor implementación del Diffserv para superar los problemas de calidad de servicio en las redes IP.

c).- Buscar mecanismos de reducción de costos para la implementación del MPLS con TE para la atención de la mayoría de los usuarios, brindándoles una mejor calidad de servicio.

d).- Para investigaciones futuras en la línea de esta investigación quedan algunas alternativas que utilizan el campo de la prioridad **IP** para determinar cual es el tratamiento a darse al tráfico en curso:

1. **QoS DiffServ** a través de **MPLS** con *frame relay* y **OSPF** en el enlace con **WFQ** (*Weighted Fair Queueing*).
2. **QoS DiffServ** a través de **MPLS** con *frame relay* y **OSPF** en el enlace con **DWRED** (*Distributed Weighted Random Early Detection*).
3. **QoS DiffServ** a través de **MPLS** con *frame relay* y **OSPF** en el enlace con **CAR** (*Committed Access Rate*).
4. **QoS DiffServ** a través de **MPLS** con otros protocolos en el enlace y otros mecanismos.

e).- También están disponibles otras alternativas, para la selección de las clases de tráfico y parámetros de control, que pueden ser utilizadas con los mecanismos arriba indicados o con otros.

REFERENCIALES

[1] ABOUL-MAGD, O. Y JAMOUSSI, B. QoS and service interworking using constraint-route label distribution protocol (CR-LDP). IEEE Communications Magazine - Nortel Networks, 2001.

[2] ABREU, P. R. Y RAUNHEITTE, L. T. M. Análisis de desempeño da tecnología frame relay sobre a tecnología ATM. São Paulo. Universidade Presbiteriana Mackenzie, 2001.

[3] ALAM, M.; PRASAD, R. Y FARASEROTU, J. R. Quality of service among IP - based heterogeneous networks. IEEE, 2001.

[4] ALMQUIST, P. Type of service in the internet protocol suite. IETF RFC 1349 - Network Working Group, 1992.

[5] ANDERSSON, L. LDP specification. IETF RFC 3036, 2001.

[6] AWDUCHE y Otros. Requirements for traffic engineering over MPLS. Disponible en: http://www.draft-ietf-mpls-traffic-eng-01.txt.

[7] Bayle, T.; Aibara, R. e Nishimura, K. Performance measurements of MPLS traffic engineering QoS. [S.l.]: IEEE Symposium on Computers and Communications. p. 5-8, 2001.

[8] BERSON, S. RSVP protocol overview. White Paper, 1999.

[9] BLAKE, S. y Otros. An architecture for differentiated services. Disponible en: http://www.faqs.org/rfcs/rfc2475.html.

[10] BRADEN, R. T. Requirements for Internet Hosts Communications Layers. IETF RFC 1122 - Network Working Group, 1989.

[11] Cisco Labs Tests. TF_TANT MPLS phase 2 testing. [S.l.: s.n.]. Não paginado, 2000. Disponível em: <www.crihan.fr/mpls/mpls.html>.

[12] Cisco Systems. Quality of service overview. [S.l.]: User Manual. p. 1-12, 1999.

[13] Cisco Systems. A comparison between IPsec and multiprotocol label switching virtual private networks. [S.l.]: White Paper. p. 1-5, 2000.

[14] Cisco Systems. Configuring basic MPLS using OSPF. [S.l.]: User Manual. No paginado, 2001.

[15] Cisco Systems. MPLS concepts. [S.l.]: User Manual. p. 1-53, 2001.

[16] Cisco Systems. Cisco any transport over multiprotocol label switching frequently asked question. [S.l.]: White Paper. No paginado, 2002.

[17] Cisco Systems. Configuring MPLS basic VPN with RIP on customer side. [S.l.]: User Manual. No paginado, 2002.

[18] Cisco Systems. Multiprotocol label switching troubleshooting. [S.l.]: User Manual. No paginado, 2002.

[19] Cisco Systems. Subnet zero and all-ones. [S.l.]: User Manual. p. 1-4, 2002.

[20] Cisco White Paper. Any transport over MPLS. [S.l.: s.n.]. p. 1-3, 2002.

[21] Cisco White Paper. Cisco IOS MPLS quality of service. [S.l.: s.n.]. p. 1-4, 2002.

[22] DARPA INTERNET PROGRAM. Internet Protocol. Arlington, Virginia. IETF RFC 791, 1981.

[23] Faucheur, F. et al. MPLS support of differentiated services. [S.l.]: IETF Internet Draft. No paginado, 2001. Disponível em: <draft-ietf-mpls-diff-ext-08.txt>.

[24] GHOSH, D.; SARANGAN, V. y ACHARYA, R. Quality of service routing in IP networks. IEEE, 2001.

[25] HOBBY, R. Performance measurement architecture version 0.2, 2002.

[26] JAMOUSSI, B. Constraint-based LSP setup using LDP. Disponible en: http://www.draft-ietf-mpls-cr-ldp-05.txt.

[27] Jupiter Networks. Supporting differentiated service classes in large IP networks [S.l.]: White Paper. No paginado, 2001.

[28] JUNIPER NETWORKS, INC. Junos internet software configuration policy framework. USA, 2003.

[29] KATZ, D. IP router alert option (RFC 2113). Cisco Systems. San Jose, CA, USA, 1997.

[30] KUROSE, J. F. y ROSS, K. Computer Networking. Addison Wesley. 2 ed., 2002.

[31] MA, T. y SHI, B. Bringing quality control to IP QoS. IETF, 2000.

[32] MARIN, P. S. Interferencia entre canales de datos y voz en el mismo cable en sistemas de cableado estruturado. 2000.

[33] MAXEMCHUK, N.F. y LOW, S.H. Active routing. IEEE. p. 1-15, 2001.

[34] MUNZNER, T. y Otros. Visualizing the global topology of the MBone. San Francisco, CA: [s.n.]. p. 1-3, 1996.

[35] Memotec Communications Incorporated. Voz/fax over frame relay vs. voz/fax over IP. [S.l]: White Paper. p. 1-20. 1998.

[36] Memotec Communications Incorporated. Quality of service in integrated voz, video, and data networks. [S.l.]: White Paper. p. 1-6, 2001.

[37] NIL Data Communications. Advanced MPLS traffic engineering - lab solutions. [S.l]: White Paper – Eslovenia. No paginado, 2001.

[38] NIL Data Communications. Configuring MPLS traffic engineering in MPLS VPN environment. [S.l.]: White Paper – Eslovenia. No paginado, 2001.

[39] NIL Data Communications. Configuring MPLS traffic engineering in MPLS VPN environment - lab solutions. [S.l.]: White Paper – Eslovenia. No paginado, 2001.

[40] NIL Data Communications. MPLS traffic engineering. [S.l.]: White Paper. No paginado, 2001.

[41] NIL Data Communications. MPLS traffic engineering lab solutions [S.l.]: White Paper. No paginado, 2001.

[42] NETPERF. A Network performance Network. USA: [s.n.]. No paginado, 2002.

[43] Nichols, K., et al. Definition of the differentiated services field (DSField) in the IPv4 and IPv6 headers. [S.l.]: IETF RFC 2474. No paginado, 1998. Disponible en: <http://www.faqs.org/rfcs/rfc2474.html>.

[44] QoS Forum. QoS protocols & arquitetures. [S.l.]: White Paper. No paginado, 1999.

[45] River Stone Networking White Paper. MPLS: making the most of Ethernet in the metro. Santa Clara - CA – USA: [s.n.]. p. 1-8, 2001.

[46] Rosen, E., et al. MPLS label stack encoding. [S.l.]: IETF RFC 3032. No paginado, 2001.

[47] Rosen, E.; Viswanathan, A. e Callon, R. Multiprotocol label switching architecture. [S.l.]: IETF RFC 3031. p. 1-54. (2001). Disponible en: <http://www.faqs.org/rfcs/rfc3031.html>.

[48] Rozolem, M. Sobre la transmisión de voz en redes IP con conmutación de rótulos. Sao Paulo: Universidad Presbiteriana Mackenzie. No paginado, 2002.

[49] Segal, B. A short history of internet protocols at CERN. [S.l.: s.n.]. No paginado, 1995.

[50] Semeria, C. e Stewart III, J. W. Supporting differentiated service classes in large IP networks. [S.l.: s.n.]. No paginado, 2001.

[51] Shaikh, F. A., et al. End - to - end testing of IP QoS mechanisms. [S.l.]: IEEE. p. 80-87, 2002.

[52] Swallow, G., et al. Multiprotocol label switching (MPLS). [S.l.]: IETF. p. 1-4, 2002.

[53] Tanembaum, A. S. Computer networks. New Jersey: Prentice Hall PTR. 3.ed. No paginado, 1996.

[54] Teltelman, B. e Hanss, T. QoS requirements for internet2 (draft). [S.l.: s.n.]. p. 552-565, 1998.

[55] Wortham, D.; Lawrence, J. e Redford, R. Positioning and developing a migration strategy to offer advanced IP services based on MPLS. [S.l.: s.n.]. Não paginado, 2000.

[56] Xiao, X. Providing quality of service the internet. Michigan: Michigan State University - Tesis. No paginado, 2000.

ANEXOS

1. Matriz de consistencia

IMPLEMENTACIÓN DE CALIDAD DE SERVICIO VÍA *MULTIPROTOCOL LABEL SWITCHING* (MPLS)				
PROBLEMAS	**OBJETIVOS**	**HIPÓTESIS**	**VARIABLES**	**MÉTODOS**
General La necesidad de implementar mecanismos de QoS en la *Internet*. **Específicos** Calidad de servicio principalmente para aplicaciones Multimedia. Entre elas, prioritariamente para	**General** Contribuir con la mejora del cuadro de QoS en la *Internet*. **Específicos** Obtener conocimiento detallado sobre diversos tipos de estructuras que implementan QoS sobre redes de	**La hipótesis** asumida en este trabajo, presupone que los recursos y herramientas de prueba facilitaran y mejoraran el bajo índice de oferta de calidad de servicio en la *Internet* beneficiando entre ellas a las aplicacione	**Variables independientes** = IPDT, IPDV, IPLR IPDT = retraso IPDV = *Jitter* IPLR = pérdida de paquetes **Variable dependiente** QoS = f (IPDT, IPDV, IPLR) QoS = Calidad de servicio	**General** Las experiencias previstas permitirán adquirir "experiencia" y profundidad en los conceptos y estadísticas, evaluando cómo las métricas se comportan cuando actúan solas o en conjunto. **Específico**

VoIP.	telecomunicaciones industriales y académicas.	s avanzadas o Multimedia, en particular a las aplicaciones del área médica, como telemedicina, telediagnóstico, etc.		Particularmente será evaluada la carga de la red en las situaciones de perdida de paquetes, visando reducir los efectos de situaciones de congestionamiento que puedan presentarse.