Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica

TRABAJO DE DIPLOMA

"Calidad de Servicio para redes IP/MPLS en Cisco"

Autor: Jans Pérez Quintero

Tutor: Ing. Rafael E. Viego Escandell

Santa Clara

2009

"Año del 50 Aniversario del Triunfo de la Revolución"

Universidad Central “Marta Abreu” de Las Villas.

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica

TRABAJO DE DIPLOMA

“Calidad de Servicio para redes IP/MPLS en Cisco”

Autor: Jans Pérez Quintero

E-mail: jpquintero@uclv.edu.cu

Tutor: Rafael E. Viego Escandell

Subgerente Datos Etecsa Villa Clara

E-mail: rviego@etecsa.cu

Santa Clara

2009

"Año del 50 Aniversario del Triunfo de la Revolución"

Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Telecomunicaciones y Electrónica, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Autor

Firma del Jefe de Departamento donde se defiende el trabajo

Firma del Responsable de Información Científico-Técnica

PENSAMIENTO

“Saber es poner en práctica lo que se sabe, y cuando no se sabe, no tratar de aparecer como que se sabe. Eso es saber”

Confucio

DEDICATORIA

A mi novia Beatriz, a mi hermana y a mi madre por tener tanta paciencia conmigo, por brindarme su apoyo incondicional y ser las luces que guían mi camino.

AGRADECIMIENTOS

- *A mis compañeros de aula y amigos de toda la vida.*
- *A mis suegros y mí cuñada Anabel por apoyarme tanto.*
- *A mi tutor Rafael por darme tanta paciencia, y demostrarme el objetivo de la vida.*
- *A toda mi familia que de una forma u otra han contribuido en este esfuerzo.*
- *A todas las personas que han contribuido cabalmente en el transcurso de mi carrera y que han brindado todo su cariño, ayuda y confianza hacia mí.*

TAREA TÉCNICA

1. Revisión bibliográfica del estado del arte en cuanto a Calidad de Servicio (QoS) en las redes de datos.

2. Análisis de las características de Calidad de Servicio en las redes MPLS.

3. Estudio detallado del funcionamiento y configuración de QoS para redes IP/MPLS en Cisco.

4. Análisis de los distintas alternativas de diseño para implementar QoS en redes IP/MPLS en Cisco.

5. Creación de un modelo de configuración para la implementación de QoS en un backbone IP/MPLS.

6. Obtención del informe del trabajo de diploma.

Firma del Autor	Firma del Tutor

RESUMEN

El presente trabajo proporciona un análisis de la tecnología de Calidad de Servicio (QoS) para redes IP/MPLS y su aplicación en los productos Cisco. Al comienzo del trabajo se aborda el término QoS, las diferentes arquitecturas de este para redes IP y como se aplican a MPLS. Varias secciones explican las ventajas de MPLS para brindar Servicios Diferenciados (DiffServ), incluyendo detalles de las etiquetas E-LSP y L-LSP, así como los modelos de túnel de DiffServ. Se resumen los mecanismos de dirección del tráfico con una mirada detallada a la vigilancia, formación del tráfico, dirección de cola activa, etcétera. El segundo capítulo cubre el modelo de funcionamiento de QoS de Cisco y el Comando de Interfaz de Línea Modular de QoS (MQC). Este capítulo explica el modelo abstracto de QoS en Cisco y proporciona una referencia completa de los comandos de configuración de Cisco IOS y Cisco IOS XR. El tercer capítulo trata las alternativas de diseño para implementar DiffServ en un backbone IP/MPLS mediante la planificación de la capacidad por clases según las exigencias de funcionamiento del backbone. Destacando que se obtienen resultados favorables en cuanto a utilización y rendimiento del ancho de la banda de la red, permitiendo aplicaciones futuras.

TABLA DE CONTENIDOS

INTRODUCCIÓN

Internet ha ido creciendo de una forma exponencial y ganando popularidad desde que se empezó su comercialización, hasta convertirse hoy en día en una herramienta casi indispensable para las empresas e incluso para los usuarios comunes. Esta evolución en las aplicaciones y usos de la red ha implicado variaciones en la arquitectura y los protocolos de la misma. Los operadores de todo el mundo están continuamente acondicionando sus redes para hacer converger los servicios tradicionales sobre el protocolo IP, dando así respuesta a la creciente demanda de sus clientes de mayor ancho de banda y reducción de los costos. Se trabajó durante mucho tiempo con un protocolo básico IP, al cuál se le han realizado variaciones y modernizaciones como lo son IP sobre ATM, IPv4, IPv6, que han tratado de cubrir las necesidades de la red conforme han ido surgiendo.

Con los servicios actuales que ofrecen en línea los operadores (videoconferencia, telefonía, televisión, entre otros) el tema de la calidad en el servicio ha tomado suma importancia, pues la retransmisión de paquetes implementada por el TCP/IP ya no es una opción válida, puesto que al tratarse de servicios más interactivos, servicios de tiempo real, las pérdidas y los retardos deben ser minimizados al máximo.

La Calidad de Servicio (QoS) permite a los administradores de redes el uso eficiente de los recursos de sus redes con la ventaja de garantizar que se asignen más recursos a aplicaciones que así lo necesiten, sin arriesgar el desempeño de las demás aplicaciones. Los esfuerzos de la industria por conseguir convergencia han generado la necesidad de un aumento de niveles de diferenciación del tráfico. Una serie de exigencias de QoS tiene que ser encontrada para soportar distintas aplicaciones (por ejemplo: datos, voz, y video) y múltiples servicios de red (por ejemplo: IP, Ethernet, ATM) en una sola red multiservicio.

El término Protocolo Múltiple de Conmutación de Etiquetas (MPLS, Multi-Protocol Label Switching), constituye un estándar de la Fuerza de Trabajo de Ingeniería de Internet (IETF, Internet Engineering Task Force) que surgió a mediados de la década del 90 para lograr un consenso entre diferentes soluciones de conmutación IP. MPLS reduce significativamente el procesamiento de paquetes que se requiere cada vez que un paquete ingresa a un enrutador de la red, mejorando el desempeño de dichos dispositivos y el desempeño de la red en general. Los routers MPLS pueden trabajar con routers IP a la par, lo que facilita la introducción de dicha tecnología a redes ya existentes como ATM y Frame Relay. Soluciona los problemas que presentan las redes IP sobre las redes ATM convencionales, tales como la expansión sobre una topología virtual superpuesta y la complejidad en la gestión de dos redes separadas y tecnológicamente diferentes, al combinar la inteligencia del enrutamiento con la rapidez de la conmutación.

MPLS ofrece nuevas posibilidades en la gestión del backbone, y permite nuevos servicios de valor añadido, así como la gestión de diferentes niveles de servicios con una mayor fiabilidad y con las garantías necesarias.

La configuración de MPLS en los productos Cisco mediante el modelo de funcionamiento de QoS de Cisco y el Comando de Interfaz de Línea Modular de QoS (MQC), ofrecen a los administradores de redes una poderosa herramienta para el diseño, despliegue, y realización de QoS en sus redes IP/MPLS.

En nuestro país hoy en día, la implementación de MPLS en la Red de Datos de ETECSA (Empresa de Telecomunicaciones de Cuba), permitirá no sólo aumentar las velocidades de conmutación sin necesidad de costosas actualizaciones de la tecnología con que cuenta el país, sino que, gracias a sus facilidades a la hora de implementar Calidad de Servicio, la red podrá incorporar el tráfico de voz junto al de datos. Por lo tanto se hace imprescindible el conocimiento de la tecnología de QoS detrás de MPLS junto con las diferentes opciones de diseño disponibles para construir un modelo de red multiservicio con productos Cisco.

El presente trabajo propone la implementación de la herramienta de Calidad de Servicio sobre redes IP/MPLS en Cisco, para lo cual se trazaron los siguientes objetivos:

1. Definir los objetivos generales de la Calidad de Servicio.
2. Analizar las principales variantes para realizar Calidad de Servicio en redes IP/MPLS.

3. Realizar una descripción del funcionamiento y configuración de la Calidad de Servicio en Cisco.

4. Proponer un modelo de configuración para realizar QoS en un backbone IP/MPLS.

Para la realización de este trabajo se plantearon las siguientes tareas investigativas:

1. Revisión bibliográfica del estado del arte en cuanto a Calidad de Servicio en las redes de datos.

2. Análisis de las características de la Calidad de Servicio en las redes MPLS.

3. Estudio detallado del funcionamiento y configuración de QoS en Cisco.

4. Análisis de los distintas alternativas de diseño para implementar QoS en redes IP/MPLS en Cisco.

5. Creación de un modelo de configuración para la implementación de QoS en un backbone IP/MPLS.

6. Obtención del informe del trabajo de diploma.

Organización del informe

El informe de la investigación se estructurará en resumen, introducción, capitulario, conclusiones y recomendaciones, glosario de términos, referencias bibliográficas y anexos.

Introducción Se dejará definida la importancia, actualidad y necesidad del tema que se aborda.

Capítulo 1 Se analizan las generalidades de la tecnología de Calidad de Servicio y el aporte de la red MPLS.

Capítulo 2 Se exponen los detalles de funcionamiento y configuración de QoS en Cisco.

Capítulo 3 En este capítulo se propone la configuración de QoS en un backbone IP/MPLS.

Conclusiones Se realizará un análisis de los resultados obtenidos a partir de los objetivos que se trazaron inicialmente.

Recomendaciones Se harán recomendaciones que tengan como objetivo enriquecer el proceso investigativo a partir de los resultados obtenidos

Referencias bibliográficas Se hará un listado de las referencias bibliográficas consultadas siguiendo la metodología existente para este fin.

Anexos Temas que no fueron abordados en el capitulario y que su importancia exija su aparición en el trabajo.

CAPÍTULO 1. Calidad de Servicio. MPLS DiffServ

Internet ha ido evolucionando con el tiempo, y los fines para los que fue creado en un principio no son los mismos de la actualidad. Al principio, IP [RFC791] fue especificado como un protocolo de mejor esfuerzo. Una de las implicaciones de esta definición de servicio era que la red intentaría entregar el tráfico a su destino en el tiempo más corto posible. Sin embargo, la red no proporcionaría ninguna garantía del alcance de ello. Esta definición de servicio estuvo acertada durante los años tempranos de Internet, cuando las aplicaciones de datos constituyeron el bulto del tráfico de Internet. Generalmente, estas aplicaciones usaron TCP y por lo tanto se adaptaron elegantemente a variaciones en amplitud de banda, pérdidas, retardos y jitter. El alcance creciente y la capacidad de Internet hicieron una infraestructura atractiva para apoyar un número creciente de aplicaciones. Además, las corporaciones, los gobiernos, y las instituciones educativas, entre otros, encontraron en el protocolo IP una opción atractiva para construir sus redes de datos privadas. Muchas de las nuevas aplicaciones IP (por ejemplo, voz y video) tenían una naturaleza de tiempo real y limitaron la tolerancia con variaciones en amplitud de banda, pérdidas, retardos y jitter [2]. Las expectativas de servicio de los usuarios de la red y sus exigencias de aplicación, hicieron insuficiente la definición del servicio de mejor esfuerzo [1].

La definición de una arquitectura QoS comenzó en medio de los años 1990. Desde entonces, el IETF ha definido dos arquitecturas de QoS para Servicios IP: Servicios Integrados (IntServ) y Servicios Diferenciados (DiffServ). La arquitectura IntServ era la solución propuesta de la inicial. Posteriormente, la arquitectura DiffServ cobró vida. El MPLS más tarde incorporó nuevas ventajas a la arquitectura DiffServ, que el IETF había definido exclusivamente para IP [3-5].

1.1 Definición de Calidad de Servicio (QoS)

En la Rec. UIT-T E800 se define la Calidad de Servicio (QoS) como **"**el efecto colectivo del rendimiento de un servicio que determina el grado de satisfacción del usuario de dicho servicio". La QoS tiene, básicamente, cuatro variantes estrechamente relacionadas: la QoS que el usuario desea, la que el proveedor ofrece, la que el proveedor consigue realmente y la que, finalmente, percibe el usuario. Los parámetros técnicos de estas redes atendiendo a los requisitos de los usuarios fueron definidos desde 1994 por el ETSI en el ETR-003, *"General Aspects of Quality of Service* (QoS) *and Network Performance* (NP)*"* [6].

1.2 Parámetros de Calidad de Servicio

Existen muchos términos para calidad de servicio, los más importantes son:

Tiempo

● Retardo: tiempo que consume un mensaje en ser transmitido y alcanzar su destino

● Tiempo de Respuesta: es el tiempo total desde que se solicitó la transmisión hasta que se recibe la respuesta.

● Variaciones del Retardo o Jitter: variabilidad en el retardo o del tiempo de respuesta.

Ancho de Banda

● Razón de Datos a Nivel de Sistema: es el ancho de banda requerido o disponible en Bits o Bytes por segundo.

● Razón de Datos a Nivel de Aplicación: es el ancho de banda requerido o disponible en unidades específicas del nivel de aplicación.

● Razón de Datos de Transacción: número de operaciones solicitadas o procesadas por segundo.

Fiabilidad

● Tiempo Medio Hacia Fallas (MTTF): es el tiempo de operación normal entre fallas.

● Tiempo Medio de Reparación (MTTR): es el tiempo medio de baja desde la falla hasta el restablecimiento de la próxima operación.

● Tiempo Medio Entre Fallas (MTBF): MTBF = MTTF + MTTR

● Porciento de Tiempo Disponible: MTTF/ MTTF + MTTR

● Razón de Pérdida de Paquetes: Proporción del total de paquetes que fue enviado y que no llega.

1.3 Beneficios al aplicar QoS

La calidad en el servicio permite a los administradores de redes usar eficientemente los recursos disponibles con la ventaja de garantizar la asignación de más recursos a aplicaciones que así lo necesiten, sin arriesgar el desempeño de las demás aplicaciones. Por lo tanto a través de Qos [1] el administrador tiene un mayor control sobre su red con menores costos y mayor satisfacción del cliente. Internet esta siendo considerado como una nueva vía para incrementar los negocios mediante la formación y el crecimiento de intranets dentro de las empresas y extranets que permiten el comercio electrónico con otras empresas. Para quienes realizan sus negocios sobre la Web es cada vez más importante que los administradores de las redes aseguren que éstas entreguen unos niveles apropiados de calidad, confiabilidad y puntualidad en la entrega [6]. [7]

1.4 Arquitecturas para brindar QoS en redes IP

El IETF ha definido dos arquitecturas de QoS para Servicios IP: Servicios Integrados (IntServ) [8] y Servicios Diferenciados (DiffServ) [9].

1.4.1 Servicios Integrados (IntServ)

El *IntServ over Specific Link Layers* (ISSLL) definió la realización de IntServ sobre protocolos de capa de eslabón diferentes (por ejemplo, Ethernet y ATM)[8]. El mecanismo IntServ propuesto por el IETF junto con el Protocolo de Reserva de Recursos [RFC2205] permite al usuario solicitar de antemano los recursos que necesita para poder brindar QoS a aplicaciones de tiempo real, por lo que cada enrutador en el trayecto efectúa la reserva solicitada. Normalmente la reserva de recursos se realiza para una secuencia de datagramas relacionados entre si [1, 8]. [10, 11]

Para poder implementar el protocolo RSVP, los enrutadores deben incorporar los siguientes elementos:

Control de Admisión: su función es la comprobar si la red tiene los recursos suficientes para poder satisfacer la petición requerida.

Control de Vigilancia: determina si un usuario tiene los permisos adecuados para satisfacer la petición requerida.

Clasificador de paquetes: clasifica los paquetes basándose en la categoría de Calidad de Servicio a la que pertenecen.

Organizador de paquetes: organiza el envío de paquetes dentro de cada categoría.

El problema con el IntServ radica en que es poco escalable, ya que necesita mantener información en cada enrutador, y otro inconveniente es que se necesita una compleja implementación para el protocolo RSVP en los enrutadores, lo que elevaría los costos.

Protocolo RSVP

El Protocolo de Reserva de Recursos [RFC2205] trata de prevenir situaciones de congestión mediante la reserva de recursos. RSVP se utiliza por el nodo externo para solicitar a la red QoS para un flujo o conjunto de flujos, y por los nodos intermedios para entregar las solicitudes de QoS al resto de los nodos de la ruta de datos, así como establecer y mantener el estado del servicio solicitado [1]. Opera sobre el protocolo IP (IPv4 o IPv6), el cual no permite realizar reserva de recursos y establecer un circuito virtual simultáneamente. De modo que, los mensajes RSVP se envían en paralelo con los paquetes IP. Está diseñado para funcionar con cualquier protocolo de encaminamiento ya sea unidifusión (*unicasting*) o multidifusión (*multicasting*) aunque no es un protocolo de transporte ni un protocolo de encaminamiento. Realiza las reservas para flujos de datos unidireccionales, por tanto, si se quiere transmitir datos entre dos terminales en ambas direcciones, se deberá realizar una reserva para cada dirección. La reserva de recursos para ese flujo de datos es iniciada y mantenida por el receptor de ese flujo de datos. Permite distintos tipos de reservas, de modo que los usuarios de un mismo grupo multicast pueden especificar el tipo que desean, consiguiendo un uso más eficiente de los recursos de

Internet. Es un protocolo transparente para los enrutadores no RSVP. Esto es debido a que RSVP es independiente del protocolo de encaminamiento y por tanto, no existe ningún problema en los entornos mixtos donde algunos de los enrutadores no utilizan RSVP. En esos casos, los enrutadores no RSVP utilizarán una técnica de encaminamiento best-effort. [12]

1.4.2 Servicios Diferenciados (DiffServ)

Es un protocolo de QoS propuesto por IETF [RFC2475 y RFC2474] que permite distinguir diferentes clases de servicio marcando los paquetes. La arquitectura DiffServ introduce muchos términos nuevos, las [RFC2475 y RFC3260] introducen la lista completa de términos, algunos de estos son:

Domain Una red con una implementación DiffServ común (por lo general en el mismo control administrativo).

Region Un grupo de dominios DiffServ contiguos.

Egress node Último nodo cruzado por un paquete antes de dejar un dominio DiffServ.

Ingress node Primer nodo cruzado por un paquete entrando en un dominio DiffServ.

Interior node El nodo en un dominio DiffServ que no es el nodo de ingreso o de egreso.

DiffServ field Este campo corresponde a los seis bits más significativos del segundo byte en la cabecera IP (antes, octeto de Tipo de servicio (TOS) en IPv4 y octeto de Clase de Tráfico en IPv6).

Differentiated Services Code Point **(DSCP**) Un valor específico asignado al campo de DiffServ.

Behavior aggregate **(BA)** Colección de paquetes que cruzan un nodo DiffServ con mismo DSCP.

Ordered aggregate **(OA)** Un juego de BAs para el cual un nodo DiffServ debe garantizar no pedir de nuevo paquetes.

BA classifier El clasificador que selecciona paquetes basado en el DSCP.

Multifield (MF) classifier El clasificador que selecciona un paquete basado en campos múltiples en la cabecera de paquete (por ejemplo, dirección de la fuente, dirección de destino, protocolo, y puerto de protocolo).

Per-hop behavior **(PHB)** Comportamiento al pasar o servicio que una BA recibe en un nodo [13].

Per-hop behavior group Varios PHBs que son puestos en práctica simultáneamente [13].

PHB scheduling class **(PSC**) Un juego de PHBs para el cual un nodo DiffServ debe garantizar no pedir de nuevo paquetes.

Traffic profile Descripción de un modelo de tráfico con el tiempo.

Marking El ajuste del DSCP en un paquete.

Metering Medición de un perfil de tráfico con el tiempo.

Policing Descarte de paquete que se ejecuta conforme a un perfil de tráfico.

Shaping Almacenamiento intermedio de paquetes que se ejecuta conforme a un perfil de tráfico.

Service level agreement **(SLA)** Los parámetros que describen un contrato de servicio entre un dominio DiffServ y un cliente de dominio.

Traffic-conditioning specification Los parámetros que ponen en práctica una especificación de nivel de servicio.

Traffic conditioning El proceso de hacer cumplir un tráfico que condiciona especificación por funciones de control como marca, medición, vigilancia, y formación.

La arquitectura DiffServ, definida en la RFC 2475, define Clases de Servicio (CoS), llamados Agregados, y funciones de gestión de recursos de QoS con operación basada en nodos o Per-Hop (por salto). La definición de clases de servicio (CoS) incluye un BA el cual define requerimientos específicos de puesta en cola y descarte de paquetes y un OA el cual efectúa la clasificación basado sólo en requerimientos de puesta en cola y puede incluir varios valores de precedencia de descarte de paquetes. El modelo DiffServ está basado en la redefinición del significado del campo ToS de 8 bits del encabezado IP. La definición original de ToS no fue implantada ampliamente, razón por la cual ahora el campo es

dividido en un subcampo de 6 bits denominado DSCP y un subcampo de 2 bits denominado ECN[1]. El valor del campo DSCP es usado para especificar un BA (una clase), el cual es usado por los nodos con DiffServ habilitado para escoger el apropiado PHB (por ejemplo un tratamiento de servicio de cola). Han sido definidos catorce PHBs[13], incluyendo uno para Reenvío Expedito (EF), doce para Reenvío Asegurado (AF)[14] y un PHB por defecto o Best Effort. Los 12 PHB de reenvío asegurado (AF) son divididos en cuatro PSC AF, y cada uno de los PSC AF consiste de tres subcomportamientos relacionados a diferente tratamiento de descarte de paquetes[9, 14]. En resumen, el modelo DiffServ permite a la red clasificar (combinar) microflujos en agregados de flujo (BA) y ofrecer a estos agregados tratamiento diferenciado en cada nodo con DiffServ habilitado. Este tratamiento es reflejado en los mecanismos de servicio los cuales incluyen la puesta en cola y descarte de paquetes. El PHB[13] se refleja en ambos aspectos, mientras que el PSC aplica solamente a la puesta en cola [1, 9]. [15]

La Figura 1.2 muestra la relación entre el campo ToS en el encabezado IP y el campo DiffServ. Las letras indican lo siguiente: D = Delay, T = Throughput, R = Reliability, C = Cost y ECN = Explicit Congestion Notification.

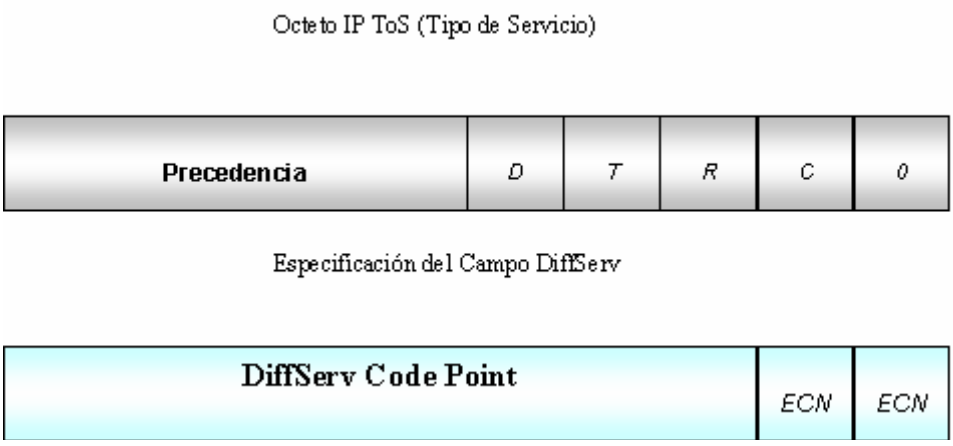

Figura 1.2 Relación entre el campo ToS en el encabezado IP y el campo DiffServ

1.5 Multi-Protocol Label Switching. Definición.

MPLS (*Multi-Protocol Label Switching*) es un estándar multiprotocolo de la IETF, RFC 3031, capaz de transportar diferentes tipos de tráfico por la misma vía usando información contenida en etiquetas en las cabeceras de los paquetes (encapsulamiento) y de encaminadores específicos capaces de reconocerlas [16, 17]. Este protocolo se basa en la asignación e intercambio de etiquetas en base a criterios de prioridad y/o calidad (Qos) [5]. MPLS es una tecnología que permite ofrecer QoS, independientemente de la red sobre la que se implemente [18]. El etiquetado en capa 2 permite ofrecer servicio multiprotocolo y ser portable sobre multitud de tecnologías de capa de enlace: ATM, Frame Relay, LANs *(Local Area Network*) [4, 16, 18, 19].

MPLS realiza las siguientes funciones:

● Especifica mecanismos para manejar flujos de tráfico entre diferentes hardware, maquinas, o incluso entre diferentes aplicaciones.

● Provee de medios para mapear direcciones IP, en etiquetas de longitud fija que son usadas por diferentes técnicas de envío y conmutación de paquetes.

● Permite el uso de diferentes protocolos existentes como son el Protocolo de Reserva de Recursos (RSVP, *Resource Reservation Protocol*) y el Primer Camino Abierto más Corto (OSPF, *Open Shortest Path First*).

● Soporta protocolos de capa 2: IP, ATM y Frame Relay.

● Permite proporcionar Servicios Diferenciados así como QoS e Ingeniería de Tráfico y crear Redes Virtuales Privadas (VPN, *Virtual Private Network*).

1.5.1 Arquitectura MPLS

Componentes:

LER (Label Edge Router): elemento que inicia o termina el túnel (pone y quita cabeceras). Es el elemento de entrada o salida a la red MPLS. Un router de entrada se conoce como *Ingress Router* y uno de salida como *Egress Router*. Ambos se suelen denominar *Edge Label Switch Router* ya que se encuentran en los extremos de la red MPLS.

LSR (*Label Switching Router*): elemento que conmuta etiquetas.

LSP (*Label Switched Path*): nombre genérico de un camino MPLS (para cierto tráfico o FEC), es decir, el túnel MPLS establecido entre los extremos. Se debe tener en cuenta que un LSP es unidireccional.

LDP (*Label Distribution Protocol*): un protocolo para la distribución de etiquetas MPLS entre los equipos de la red.

FEC (*Forwarding Equivalence Class*): nombre que se le da al tráfico que se encamina bajo una etiqueta. Subconjunto de paquetes tratados del mismo modo por el conmutador.

En la Figura 1.3 se muestran los componentes de une cabecera MPLS donde:

Label (20 bits): Es la identificación de la etiqueta.

Exp (3 bits): Llamado también bits experimentales, también aparece como CoS, afecta al encolado y descarte de paquetes.

S (1 bit): Sirve para el apilado jerárquico de etiquetas. Cuando S=0 indica que hay mas etiquetas añadidas al paquete. Cuando S=1 estamos en el fondo de la jerarquía.

TTL (8 bits): Time-to-Live, misma funcionalidad que en IP, se decrementa en cada enrutador y al llegar al valor de 0, el paquete es descartado. Generalmente sustituye el campo TTL de la cabecera IP [18].

Figura 1.3 Cabecera MPLS

1.6 Soporte de MPLS para DiffServ

El MPLS soporta DiffServ con ajustes mínimos a MPLS y la arquitectura DiffServ [20, 21]. El MPLS no introduce ninguna modificación en el acondicionamiento de tráfico y conceptos PHB definidos en DiffServ [22]. Un router de conmutación de etiquetas (LSR, *Label Switching Router*) usa los mismos mecanismos de dirección de tráfico (medición, marca, formación, vigilancia, cola, etcétera) para condicionar y poner en práctica PHBs

diferentes para el tráfico MPLS. Una red MPLS puede usar la ingeniería de tráfico para complementar su realización DiffServ. La RFC 3270 define el apoyo de MPLS a la arquitectura DiffServ. Una red de MPLS puede poner en práctica DiffServ para soportar diversas exigencias de Calidad de Servicio en una manera ampliable. El MPLS DiffServ no es específico al transporte del tráfico de IP sobre una red MPLS [21]. Una red MPLS puede llevar otros tipos del tráfico para el cual DiffServ no se aplica (por ejemplo, ATM o Frame Relay). La red MPLS puede crecer sin necesidad de introducir mayores cambios en su diseño DiffServ designando caminos de conmutación de etiquetas (LSPs, *Label Switched Paths*) a medida que la red aumenta. Estas características desempeñan un papel importante en la realización de grandes redes MPLS que pueden transportar un amplio espectro del tráfico [20]. El soporte de MPLS para DiffServ introduce dos tipos de LSPs con diferentes características de servicio y operación. El primer tipo, *EXP-inferred-class* LSP (E-LSP) puede transportar simultáneamente múltiples clases de tráfico. El segundo tipo, *Label inferred class* LSP (L-LSP), transporta una sola clase [23]. [21]

Las especificaciones para MPLS DiffServ definen dos codificaciones para el punto de código de DiffServ. Por tanto estos tipos de LSP imponen diferentes exigencias de señalización. [24]

1.6.1 El E-LSP

MPLS define el E-LSP como un tipo de LSP que puede llevar simultáneamente múltiples clases de tráfico [23]. Los LSRs usan el espacio EXP en la cabecera para deducir el PHB que un paquete requiere. El EXP contiene tres bits que pueden tomar ocho valores posibles. El tamaño de este espacio implica que un E-LSP puede transportar hasta ocho clases del servicio. Un E-LSP lleva menos clases si algunas de esas clases presentan marcas múltiples (por ejemplo, AF1 que puede usar dos o tres marcas). Las especificaciones no definen valores recomendados de EXP para los PHBs (EF, AF n, CS n por defecto) existentes. Además, ellos no definen ninguna estructura en el espacio de tres bits. Los LSRs pueden colocar E-LSPs con la reservación de ancho de banda (generalmente para el control de admisión). [23, 24]

MPLS define mecanismos para los E-LSPs con el objetivo de señalar correlaciones entre valores de EXP y PHBs. Un LSR asocia correlaciones EXP→PHB para etiquetas de

entrada y correlaciones PHB→EXP para etiquetas de salida. La señalización es opcional y ocurre durante el sistema LSP. La RFC 3270 define extensiones a LDP (DiffServ TLV [Tipo, Longitud, Valor para Petición de Etiqueta], Correlación de Etiqueta, Liberación de Etiqueta, y Notificación de mensajes) y RSVP (objeto de DiffServ para mensajes de Camino) y su procesamiento apropiado. La señalización identifica el LSP como un E-LSP y especifica las correlaciones entre los valores de EXP y PHBs que estos usarán. Los LSRs pueden usar correlaciones estáticas pero configurables para evitar estas extensiones señaladas para los E-LSPs. Un LSR debería trazar un mapa de todos los valores de EXP a PHB por defecto si la señalización de LSP no especificara una correlación y no existiera ninguna correlación preconfigurada. [23]

1.6.2 El L-LSP

MPLS define el L-LSP como un tipo de LSP que puede transportar una sola clase de tráfico. Los LSRs deducen la clase asociada con un paquete de la etiqueta y determinan el PHB exacto utilizando la etiqueta en combinación con el campo EXP. La Tabla 1.1 ilustra la correlación obligatoria entre <clases, EXP> y PHBs. Los LSRs aprenden la asociación entre etiquetas de L-LSP y clases durante el sistema LSP. Los L-LSPs requieren el uso de extensiones de señalización DiffServ. En este caso, los LSRs usan un formato diferente del LDP DiffServ TLV y RSVP DiffServ. La señalización identifica el LSP como un L-LSP y especifica la clase que el L-LSP transportará. Como con los E-LSPs, los LSRs pueden establecer L-LSPs con la reservación de ancho de banda. [24]

Tabla 1.1 Correlaciones de PHB para L-LSPs

Clase	**EXP (Decimal)**	**EXP (Binario)**	**PHB**
EF	0	000	EF
AF4	3	011	AF43
AF4	2	010	AF42
AF4	1	001	AF41
AF3	3	011	AF33
AF3	1	010	AF32

AF3	3	001	AF31
AF2	2	011	AF23
AF2	1	010	AF22
AF2	3	001	AF21
AF1	2	011	AF13
AF1	1	010	AF12
AF1	1	001	AF11
CS7	0	000	CS7
CS6	0	000	CS6
CS5	0	000	CS5
CS4	0	000	CS4
CS3	0	000	CS3
CS2	0	000	CS2
CS1	0	000	CS1
Por defecto	0	000	Por defecto

El uso de E-LSPs y L-LSPs en una red de MPLS no es mutuamente exclusivo. Los LSRs mantienen un contexto de etiqueta de DiffServ. Este contexto indica el tipo de LSP (E-LSP o L-LSP), los PHBs de los LSP, y la correlación entre el paquete de encapsulación y un PHB. Para etiquetas de entrada, esta correlación define como el LSR puede deducir el PHB del paquete de encapsulación. Para etiquetas de salida, esta correlación define como el LSR codifica el PHB. Este contexto es poblado con correlaciones preconfiguradas o por la información DiffServ aprendida durante la configuración del LSP. [20, 24]

La Figura 1.4 muestra una red MPLS usando L-LSPs y E-LSPs simultáneamente. En este ejemplo, hay dos E-LSPs entre nodo E y nodo D, y dos L-LSPs entre el nodo A y D. La red soporta tres clases: EF, AF1, y AF2. En este ejemplo, el nodo C transporta tanto E-LSPs

como L-LSPs. Este nodo usa el contexto de la etiqueta DiffServ para determinar el tipo de LSP y la correlación exacta que se debe usar para deducir el PHB del paquete de encapsulación. Los LSRs entregan los paquetes según su PHB sin tener en cuenta el LSP y su tipo. Los detalles de LSP influyen en la determinación PHB, pero el PHB por último determina el tratamiento de paquete. En este ejemplo, los nodos A y E usan un tipo de LSPs exclusivamente. Sin embargo, cada uno de ellos podría usar una combinación de E-LSPs y L-LSPs para alcanzar el nodo D.

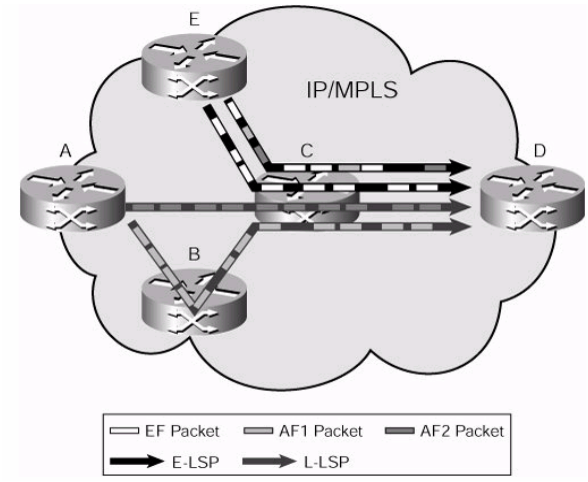

Figura 1.4 Red combinando L-LSPs y E-LSPs

1.6.3 Modelos de túnel de MPLS para DiffServ

La especificación para del soporte de MPLS para DiffServ define tres modos diferentes de la operación con reglas específicas de interacción entre la marca DiffServ en la cabecera IP y la marca DiffServ en la cabecera MPLS. Estos tres modelos de túnel son llamados: Uniforme, Tubo, y Tubo Corto. Usando estos modelos de túnel, una red de MPLS puede construir un túnel definiendo un dominio DiffServ separado o la red MPLS puede actuar como parte de un mayor dominio de DiffServ. Los tres modelos no introducen ningún cambio en la conmutación de un LSR o alguna señalización requerida. Estos modelos se aplican igualmente para E-LSPs y L-LSPs. Los modelos de tunelado de DiffServ sobre MPLS amplían los conceptos introducidos en la RFC 2983. Esta RFC define la operación de DiffServ sobre túneles IP. En muchos aspectos, los LSPs se parecen a túneles IP. La RFC 3270 define los modelos de tunelado de DiffServ sobre MPLS para el transporte de IP

y tráfico MPLS. Sin embargo, los conceptos subyacentes pueden aplicarse al transporte de otros tipos del tráfico (por ejemplo, Ethernet) sobre una red de MPLS. [4, 24]

Modelo Uniforme

En el modelo de túnel, mostrado en la Figura 1.5, hay sólo una marca de DiffServ que es relevante para un paquete cruzando la red de MPLS. Si la marca de DiffServ del paquete es modificada dentro de la red de MPLS (debido al acondicionamiento de tráfico, etc.) la información actualizada es de un significativo valor en el egreso del LSP. Cualquier cambio en la marca de paquete dentro de la red de MPLS es permanente y es propagada cuando el paquete deja la red MPLS. [24]

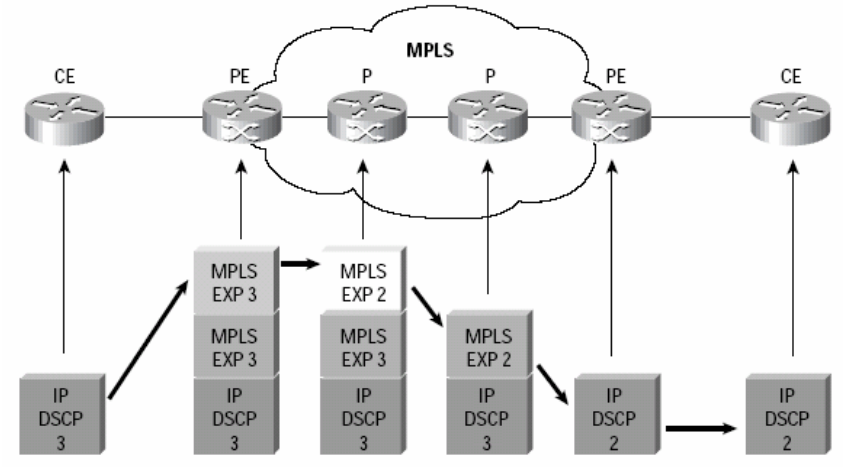

Figura 1.5 Modelo de túnel uniforme

Modelo de Tubo

En este modelo de túnel, mostrado en la Figura 1.6, dos marcas son relevantes para un paquete cruzando la red MPLS. Primero, la marca usada por los nodos intermedios a lo largo de la travesía del LSP incluyendo el LSR de egreso. Segundo, la marca inicial llevada por el paquete antes de entrar a la red de MPLS que seguirá siendo usada una vez el paquete deja la red MPLS. Cualquier cambio en la marca de paquete dentro de la red MPLS no es permanente y no se propaga cuando el paquete deja la red MPLS. Se puede ver que el

LSR de egreso todavía usa la marca que fue usada por los LSRs intermedios. Sin embargo, el LSR de egreso tiene que borrar todas las etiquetas impuestas al paquete original. A fin de conservar esta marca llevada en las etiquetas, el LSR de borde guarda una copia interna de la marca antes de borrar las etiquetas. Esta copia interna es usada para clasificar el paquete en la interfaz de salida una vez que las etiquetas son borradas. [24]

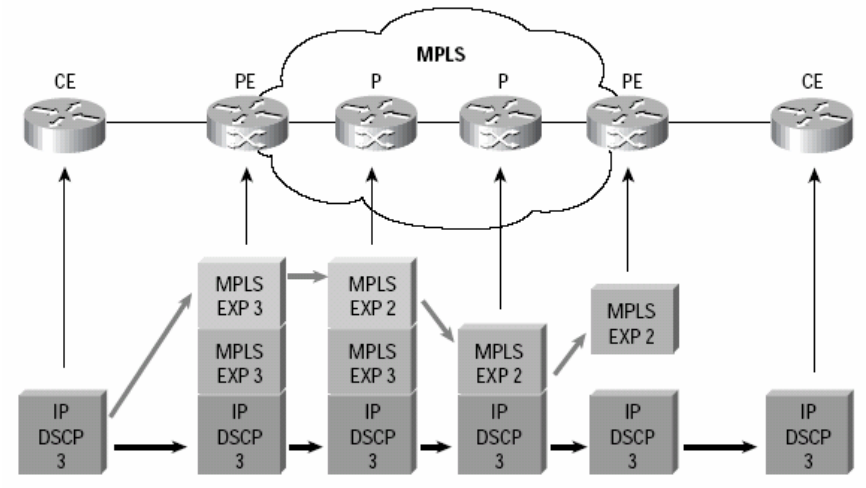

Figura 1.6 Modelo de túnel de tubo

Modelo de Tubo Corto

El modelo de túnel de Tubo corto, mostrado en la Figura 1.7, es una variación leve del modelo de túnel de Tubo. La única diferencia es que el LSR de egreso usa la marca del paquete original en vez de usar la marca usada por los LSRs intermedios. [24, 25]

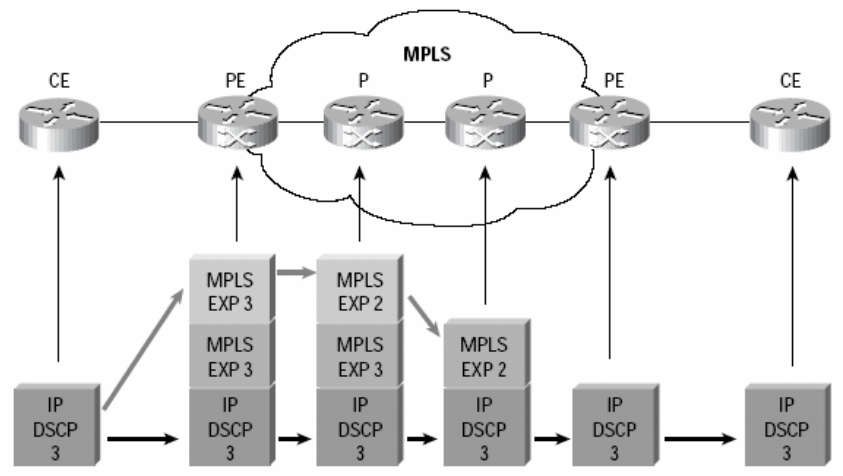

Figura 1.7 Modelo de túnel de Tubo Corto

1.7 Mecanismos de Dirección del tráfico

La realización de Calidad de Servicio depende de una serie de mecanismos de dirección de tráfico. Estos mecanismos ayudan a los nodos de la red a evitar y manejar la congestión. Estos mecanismos representan los componentes básicos que conectan a la red el uso de nodos para poner en práctica la Calidad de Servicio usando tanto el DiffServ como arquitecturas IntServ. A continuación se resumen los conceptos claves detrás de la dirección de tráfico dentro del contexto de redes IP/MPLS. [25, 26]

Clasificación del Tráfico

Los nodos de red generalmente realizan la clasificación de tráfico antes de aplicar mecanismos de dirección de tráfico. El tráfico agregado que cruza un nodo combina el tráfico con exigencias de Calidad de Servicio diferentes. En aquellos casos, los nodos de la red tienen que clasificar el tráfico para proporcionar el nivel esperado de la diferenciación. La clasificación de tráfico puede estar en la forma de un MF o BA en el contexto de DiffServ [24]. [26]

Marca del Tráfico

La marca de paquete implica asignar un nuevo valor a un espacio relacionado con la Calidad de Servicio en el encabezamiento de un paquete. Esta marca asocia el paquete con una clase o una prioridad de salida. La arquitectura DiffServ depende de la marca del paquete para indicar el PHB [13] para cada paquete [24]. [26]

Vigilancia del Tráfico

En situaciones diferentes, un nodo de red podría tener que controlar la cantidad de un flujo de tráfico particular. Un *policer* (vigilante) mide el tráfico y compara la medida con un perfil de tráfico predefinido. El resultado de comparación determina la acción que el *policer* toma con el paquete. Las tres acciones principales son: transmisión, marca, o caída (*dropping*) del paquete. La vigilancia es esencial para el acondicionamiento del tráfico en DiffServ. No es exclusivo a esta arquitectura, muchas tecnologías hacen el uso de la vigilancia de tráfico (por ejemplo, ATM y Frame Relay). En general, la vigilancia de tráfico es un mecanismo popular en límites entre dominios administrativos. [25, 26]

Formación del tráfico

La formación es otro mecanismo comúnmente usado para el control del tráfico. Similar a un *policer*, un *shaper* (formador) mide el tráfico y compara la medida con un perfil. En este caso, el resultado de comparación determina si el *shaper* debería retrasar el paquete o permitir que continúe el procesamiento. Por lo tanto, la formación requiere almacenamiento intermedio o hacer cola con los paquetes que exceden el perfil. La formación es también esencial al acondicionamiento de tráfico en DiffServ. [25]

Control de la Congestión

La asignación de memoria y la programación del tráfico son dos mecanismos que se utilizan para la dirección de la congestión. Cuando la congestión ocurre, el interfaz maneja el exceso del tráfico por almacenamiento o haciendo cola para el tráfico. Un nodo puede

crear colas múltiples en un punto de congestión dado. Cada cola puede recibir una asignación diferente de memorias intermedias y ancho de banda. Esta asignación de recurso, junto con la disciplina de programación entre colas, proporciona diferentes latencias, jitter, y características de pérdida para el tráfico en las diferentes colas. [25]

Una variante en la dirección de la congestión usa una cola simple con el principio FIFO (primero en entrar, primero en salir). En una cola con FIFO, un interfaz transmite los paquetes en el orden en que ellos llegan. Los nuevos paquetes son añadidos al final de la cola. La cola crece mientras los paquetes llegan más rápido que lo que el interfaz puede enviarlos. En algún punto, la cola podría exceder la capacidad de almacenamiento. Una política común es dejar caer el paquete, favoreciendo a los que ya se encontraban en cola. Esta política recibe el nombre de gota o caída de cola. Otro planificador es el *Weighted fair queuing* (WFQ). El WFQ computa el tiempo de salida del paquete como si un planificador de bit por bit de retorno al punto de origen estuviera en uso. El planificador selecciona paquetes para la transmisión en el orden de sus tiempos de salida. [25, 26]

Dirección de Cola Activa

Los nodos de la red tienen que manejar las colas para controlar su longitud. La dirección de cola activa [RFC2309] consiste en la caída (gota) o marca de paquetes antes de que una cola se llene. La gota y en algunos casos la marca, desempeñan un papel importante en la señalización de la congestión para fuentes TCP, que consumen la mayor parte de el ancho de banda en las redes. La forma más simple de la dirección de cola es deja caer los nuevos paquetes que llegan a una cola llena. Esta propuesta, usa un tamaño de cola máximo fijo. La gota proporciona la diferenciación limitada de paquetes y puede tener mejores efectos en el control de congestión de TCP. [25, 26]

Fragmentación e Intercalado del Enlace

Un nodo de la red puede requerir realizar el enlace fragmentado e intercalado para reducir la latencia para el tráfico con prioridad. Un paquete con prioridad puede llegar inmediatamente que la transmisión de un paquete sin prioridad comienza. Este paquete tendría que esperar a que la transmisión anterior culmine. Debido al tamaño y la velocidad requerida por del paquete, este podría experimentar una cantidad inaceptable de latencia. La

fragmentación de los paquetes de no prioridad grandes y el intercalado de fragmentos con paquetes de prioridad elimina la latencia para el tráfico de prioridad. [25, 26]

Compresión de la Cabecera

La compresión de la cabecera proporciona una mayor eficiencia del ancho de banda y reduce la latencia de transmisión. El tráfico de tiempo real generalmente confía en el Protocolo de Transporte de Tiempo Real (RTP). Según el tipo del tráfico de tiempo real, la información de la cabecera puede ser más grande que la carga útil de tiempo real. Para reducir el gasto, un nodo puede usar la compresión del encabezamiento RTP (cRTP), definido en la RFC2508 para comprimir IP, UDP, y encabezamientos RTP. El final distante del enlace reconstruye los encabezamientos antes de que el paquete continúe. La reducción del tamaño del paquete ahorra ancho de banda en el enlace y reduce la latencia de transmisión debido a que el nodo tiene que transmitir menos bits para pasar el paquete RTP. [25, 26]

1.8 Señalización de QoS

Las redes de IP/MPLS realizan la señalización de Calidad de Servicio usando RSVP y extensiones a otros protocolos de IP. La señalización trae el comportamiento dinámico al comportamiento de Calidad de Servicio en la red. Aunque la señalización de QoS demuestre útil en algunos casos, esta no es una exigencia para el despliegue de QoS. La definición de la arquitectura IntServ causó el desarrollo de RSVP para la reservación de recurso. El RSVP es el protocolo de opción para la señalización de QoS en redes de IP, pero es también aplicable fuera del contexto de QoS, un protocolo señalado para ingeniería de tráfico de LSPs. Además de las capacidades de RSVP, hay extensiones propuestas a otros protocolos de IP (por ejemplo, *Open Shortest Path First* [OSFP], *Border Gateway Protocol* [BGP], *Label Distribution Protocol* [LDP]) que traen alguna QoS o capacidades de señalización relacionadas con la QoS a aquellos protocolos. [25, 26]

1.9 Ingeniería de Tráfico

La ingeniería de tráfico puede ser utilizada como un complemento al mecanismo DiffServ para proporcionar una mejor utilización de los recursos de la red. MPLS es considerado estratégicamente una solución para la ingeniería de tráfico porque puede proveer más funcionalidad de manera integrada y con bajo coste. DiffServ tiene su importancia por proveer escalabilidad multiclase de servicios, y puede ser complementada por mecanismos de ingeniería de tráfico en MPLS [RFC 2702] que operan de manera agregada entre todos los PHBs DiffServ [24]. En ese caso, ambos proveen beneficios, DiffServ realiza diferenciación de servicio por salto y la ingeniería de tráfico de MPLS encuentra una mejor distribución de la carga del tráfico entre el conjunto de recursos de red [27]. [28]

1.10 Conclusiones del capítulo

La red MPLS puede crecer sin necesidad de introducir mayores cambios en su diseño DiffServ designando caminos de conmutación de etiquetas (LSPs, Label Switched Paths) a medida que la red aumenta. MPLS garantiza Calidad de Servicio, realiza Ingeniería de tráfico sobre la red, brinda flexibilidad, escalabilidad y hace un uso eficiente del ancho de banda de la red. Luego de un análisis general de la Calidad de Servicio y la tecnología MPLS se puede concluir que es una arquitectura que permite ampliar las ventajas de DiffServ.

CAPÍTULO 2. Funcionamiento de QoS en Cisco

Este capítulo proporciona una descripción de la realización y configuración de la Calidad de Servicio (QoS) en los productos de Cisco [25]. Esta descripción incluye detalles sobre comandos de configuración y algoritmos. Para una mayor explicación se incluyen ejemplos simples que ilustran el uso de estos comandos. Se muestran detalles de implementación tanto de Cisco IOS [29, 30] como de Cisco IOS XR [31]. Este capítulo no incluye plataforma o detalles de hardware debido a las diferentes especificaciones de las distintas realizaciones. [32, 33]

2.1 Modelo de Funcionamiento de QoS de Cisco

El modelo de funcionamiento de QoS de Cisco es un modelo abstracto que proporciona los detalles de la realización y facilita la realización de QoS a través de diferentes familias de productos. El modelo es bastante flexible para proporcionar una amplia variedad de comportamientos posibles a pesar de su simplicidad. [32, 34]

El modelo de funcionamiento de QoS usa el concepto de nodo de dirección de tráfico (TMN, *Traffic-Management Node*) [26]. Este concepto representa una abstracción de un conjunto de acciones que un dispositivo aplica al tráfico en un punto particular durante el transporte de paquetes. El TMN identifica uno o varios flujos de tráfico y define las acciones a realizar en cada flujo. La realización subyacente deduce que estructuras y mecanismos (incluso colas posibles) proporcionarán el comportamiento que los TMN definen. [32]

El TMN tiene cuatro componentes:

▪ Clasificación (classification)

▪ Precola (prequeuing)

▪ Formación de la cola (queuing)

▪ Postcola (post-queuing)

En la Figura 2.1 se muestra una vista funcional de un paquete al cruzar un TMN.

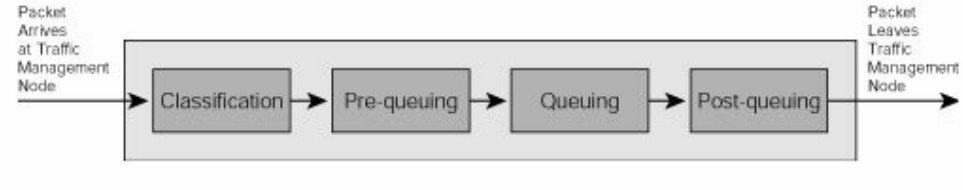

Figura 2.1 Componentes de un TMN

Clasificación:

El componente de clasificación identifica corrientes de tráfico usando el contenido del paquete o información de contexto. El TMN asocia cada corriente de tráfico con un nombre de clase. El TMN usa las cabeceras de los paquetes para clasificar el tráfico, esto incluye las cabeceras de la capa 2, capa 3, y capa 4. El componente de clasificación también puede inspeccionar la carga útil del paquete o usar la información de contexto del paquete como el interfaz de entrada. Todo el tráfico que no se corresponde con ningún criterio de clasificación explícitamente configurado se hace parte de una clase por defecto que usa el nombre de class-default. Si el componente de clasificación no existe, todo el tráfico se hace parte de esta clase por defecto. En resumen, el componente de clasificación recibe un flujo de tráfico e identifica una o varias corrientes que este asocia con nombres de clase. [26, 32]

Preformación de la cola (pre-queuing):

El componente pre-queuing agrupa un conjunto de acciones que deben preceder la formación de la cola en el TMN. Es el segundo paso en la lista del TMN, situado luego del componente classification. Este incluye acciones como vigilancia (policing), marca (marking), caída (dropping) y compresión de cabecera. El componente pre-queuing debe preceder a un componente queuing siempre que este último exista. [26, 32]

Formación de la cola (queuing):

El componente queuing dirige la asignación de ancho de banda durante los periodos de congestión. El componente queuing tiene dos subcomponentes: enqueuing y dequeuing.

El enqueuing controla el tamaño de una cola decidiendo que paquetes entran a la cola. Un ancho máximo de cola representa la forma simple del control que pone en práctica una política de gota de cola. Es decir, el enqueuing de paquetes se para cuando la cola alcanza el tamaño de cola máximo. El dequeuing controla la salida de paquetes de las colas. Cuatro atributos pueden influir en el tráfico dequeuing:

• La garantía mínima de ancho de banda representa el peor caso de asignación de ancho de banda que la cola recibirá.

• El ancho de banda de exceso define la distribución de ancho de banda que excede la garantía mínima.

• El atributo de prioridad define si el planificador debe atender una cola delante de todas otras colas de prioridad inferior. [26, 32]

Post cola (post-queuing)

El post-queuing define las acciones de QoS que deben seguir al queuing en el TMN. Este es el último componente del TMN y define el último grupo de acciones antes de que el paquete deje el TMN. Como con el pre-queuing, este componente no implica que un componente queuing deba existir. Sin embargo, este debe seguirlo si está presente. [26, 32]

2.2 Comando Modular de Interfaz de Línea Modular de QoS (MQC)

Cisco usa el Comando de Interfaz de Línea Modular de QoS (MQC, Modular QoS Command-Line Interface) como el marco de configuración para los modelos de funcionamiento de QoS. El MQC actúa como un interfaz de configuración para el TMN subyacente. El MQC facilita el despliegue de QoS suministrando un conjunto común de comandos con la misma sintaxis y semántica. Al mismo tiempo, esto proporciona plataformas de mayor flexibilidad en la selección de la implementación de QoS [30, 32]. El MQC tiene tres componentes:

▪ **Asignación de Clase (Class map)**: Define una clase de tráfico usando las reglas correspondientes. Esto corresponde al componente de clasificación del TMN.

▪ **Mapa de vigilancia (Policy map)**: Define una política que contiene acciones de Calidad de Servicio para ser aplicadas a algunas clases del tráfico. Este hace referencia a las clases definidas por los comandos *class-map* y proporciona la configuración para la pre-queuing, queuing, y post-queuing en el TMN.

▪ **Política de Servicio (Service policy)**: Asocia una política con un objetivo particular y dirección dentro de un dispositivo. [26, 32]

El Ejemplo 2.1 muestra una política de QoS usando el MQC. Este ejemplo incluye dos definiciones de clase explícitas: CLASS1 y CLASS2. La política con el nombre POLICY1 se refiere a aquellas dos clases además de la clase por defecto (class-default). Como se mencionó antes, esta clase no requiere configuración y representa todo el tráfico que no presenta clases explícitamente configuradas. La política es adjuntada al interfaz GigabitEthernet3/0 en la dirección de entrada. Por lo tanto, la política tratará paquetes que entran en el dispositivo por aquel interfaz. [30]

Ejemplo 2.1 Política de QoS usando el MQC

```
class-map match-all CLASS1
     match <statement-1>
   class-map match-any CLASS2
     match <statement-2>
     match <statement-3>
     match <statement-4>
   !
   policy-map POLICY1
     class CLASS1
      police <action-1>
     class CLASS2
      <action-2>
      <action-3>
     class class-default
      <action-4>

   !
```

```
interface GigabitEthernet3/0
 ip address 192.168.0.1 255.255.255.254
 service-policy input POLICY1
!
```

El comando *show policy-map* es el comando primario para verificar la operación y configuración de una política de QoS. Como resultado, este comando visualiza los contadores relativos a todas las acciones configuradas en la política. Los comandos *clear counters* y *clear qos counters* borran los contadores en Cisco IOS y Cisco IOS XR respectivamente. En el Tabla 1 del Anexo 1 se muestran las tres formas más comunes de el comando *show policy-map*. [30]

2.3 Mecanismos de dirección del tráfico

A continuación se muestran las opciones de configuración en el MQC presentando los comandos que permiten la configuración de los componentes del TMN (clasificación, preformación de la cola, formación de la cola, y post cola). [30]

2.3.1 Clasificación del tráfico

La clasificación del paquete se configura usando los comandos del *class map*. Los *class map* definen el componente clasificación en el TMN. Estos comandos proporcionan una amplia variedad de criterios para la clasificación del paquete. Estos criterios varían de Capa 2 (por ejemplo, dirección MAC, ATM Cell Loss Priority [CLP]) a criterios de nivel de aplicación (por ejemplo, un URL). Dentro de una política el proceso de clasificación para un paquete se termina cuando el paquete empata con una clase. Por lo tanto, la clasificación puede asociar cada paquete con una sola clase. Los paquetes que no satisfacen los criterios correspondientes de ningún *class map* forman parte de la clase por defecto. En las Tablas 2 a 6 del Anexo 1 se proporciona un resumen de la mayor parte de los criterios de correspondencia que el MQC soporta. [30]

Un *class map* soporta operaciones lógicas de los comandos *match*, pudiendo definir un OR lógico, un AND lógico, o una negación de estos comandos. El *match-any* define un OR lógico de todas las declaraciones match en el *class map*, mientras que el *match all* define un

AND lógico. En adición, el comando *match not* niega el criterio individual de emparejamiento. Algunos match (por ejemplo, *match dscp* y *match mpls experimental topmost*) aceptan lista de valores. En esos casos, un paquete satisface la declaración si este empareja cualquiera de los valores en la lista. [30]

El Ejemplo 2.2 muestra cuatro configuraciones de clases diferentes. La primera clase, CLASS1, incluye paquetes que emparejan con una lista de acceso 99 o tienen un valor EF del DSCP. CLASS2 empareja los paquetes con un valor AF11, AF12, o AF13. Los paquetes MPLS con valores del campo EXP de 3 o 4 emparejan con CLASS3. CLASS4 empareja con las celdas ATM OAM y CLASS5 con los paquetes de IPv6 con un DSCP por defecto.

Ejemplo 2.2 Configuración de la clasificación del tráfico

```
class-map match-any CLASS1
     match access-group 99
     match dscp ef
   class-map match-all CLASS2
     match dscp af11 af12 af13
   class-map match-all CLASS3
     match mpls experimental topmost 3 4
   class-map match-all CLASS4
     match atm oam
   class-map match-all CLASS5
     match protocol ipv6
     match dscp default
   !
```

2.3.2 Marca del tráfico

La marca es una de las acciones del componente pre-queuing en el TMN. El comando **set** es el método principal para marcar un campo asociado a un paquete. Este comando presenta una amplia variedad de criterios de marca, incluyendo Capa 2, Capa 3, y espacios internos. Una clase puede incluir múltiples comandos *set* para diferentes campos (por ejemplo, un comando marca el encabezamiento Capa 3, y un segundo, el encabezamiento Capa 2). Por lo tanto este comando se aplica tanto para políticas de entrada como de salida [30]. En las

Tablas 7 a 9 del Anexo 1 se proporciona un resumen de la mayor parte de los criterios de marca que el MQC soporta. Las acciones de marca por defecto se muestran en las Tablas 1 y 2 del Anexo 2.

El MQC usa un grupo de Calidad de Servicio ID y una clase de descarte como espacios internos que un dispositivo puede asociar con un paquete. El espacio del grupo de Calidad de Servicio ID representa un identificador de clase mientras que la clase de descarte corresponde a un identificador de perfil de gota. Un dispositivo puede poner estos espacios sin cambiar los contenidos del paquete. Ambos espacios usan números enteros. La información se pierde tan pronto como el dispositivo transmite el paquete por la interfaz de salida. En la mayor parte de casos, la política de entrada pone los valores y la política de salida hace el uso de ellos. [30]

Se puede usar el comando **set** para poner en práctica una correlación entre dos marcas, definiendo correlaciones entre DSCP, precedencia IP, campo EXP de MPLS, marcas internas, y la prioridad de usuario de 802.1Q. Por defecto, el comando implementa una correlación de uno a uno. Sin embargo, usted puede configurar una correlación arbitraria usando un mapa tabla como se muestra en las Tablas 10 y 11 en el Anexo 1. [30]

El Ejemplo 2.3 muestra cuatro políticas diferentes que marcan el tráfico. El POLICY1 clasifica el tráfico IP usando el campo DSCP y define el valor EXP de MPLS. El POLICY1 es válido sólo como una política de entrada. El POLICY2 clasifica paquetes MPLS usando su valor EXP y marca localmente el paquete utilizando un valor del grupo de Calidad de Servicio ID. El POLICY2 es válido sólo como una política de entrada. El POLICY3 ilustra una política con múltiples acciones que marcan todo el tráfico Ethernet 802.1Q con un valor de prioridad de usuario de 5 y un DSCP para IP de EF. Por último el POLICY4 define una correlación entre EXP de MPLS y el grupo de Calidad de Servicio ID utilizando el mapa de tabla.

Ejemplo 2.3 Políticas realizando la marca del tráfico

```
class-map match-all CLASS1
    match dscp ef
  class-map match-all CLASS2
    match mpls experimental topmost 5
```

```
class-map match-all CLASS3
  match mpls experimental topmost 3 4
!
table-map FROM-EXP-TO-QoS-GROUP
 map from 1 to 1
 map from 2 to 1
 map from 3 to 3
 map from 4 to 3
 map from 5 to 5
 default 0
!
policy-map POLICY1
  class CLASS1
   set mpls experimental imposition 5
  class class-default
   set mpls experimental imposition 0
!
policy-map POLICY2
  class CLASS2
   set qos-group 5
  class CLASS3
   set qos-group 3
  class class-default
   set qos-group 0
!
policy-map POLICY3
  class class-default
   set dscp ef
   set cos 5
!
policy-map POLICY4
 class class-default
  set qos-group mpls experimental topmost table FROM-EXP-TO-QoS-GROUP
!
```

2.3.3 Vigilancia del tráfico

El comando *police* configura la vigilancia de tráfico para medir una corriente de tráfico contra un perfil y procesa los paquetes basado en la comparación. La vigilancia es otra de las acciones del prequeuing en el TMN, por lo tanto no es la causa de la cola de los paquetes. En su forma más simple, el comando *police* define una razón límite para una clase y produce la gota del tráfico si excede ese límite. Este comando presenta un gran número de opciones y proporciona gran flexibilidad. Este siempre incluye un perfil de tráfico (Tablas 12 y 13 Anexo 1), en términos de rate (razón de datos) y/o burst (ráfagas de paquetes), y un grupo de acciones implícitamente o explícitamente especificadas (Tablas 14 y 15 Anexo 1). El comando tiene un formato de línea simple (Ejemplo 2.4) o un formato de múltiples líneas (Ejemplo 2.5). [30]

Ejemplo 2.4 Formato de línea simple para el comando police

```
policy-map POLICY1
 class class-default
  police <traffic profile> <conform-action> <exceed-action> <violate-action>
!
```

Ejemplo 2.5 Formato de múltiples líneas para el comando police

```
policy-map POLICY1
 class class-default
  police <traffic profile>
    <color-definition>
    <conform-action>
    <exceed-action>
    <violate-action>
!
```

Para configurar los colores para el reconocimiento del tráfico se utilizan los comandos *conform-color* y *exceed-color* en el formato de múltiples líneas del comando de vigilancia. Esos comandos se refieren a una clase antes definida utilizando un mapa de clase. Para el

formato de línea simple se puede definir solamente la conformación del color [30]. La Tabla 16 del Anexo 1 resume los comandos que definen los colores del tráfico.

En el Ejemplo 2.6 se muestra una configuración de la vigilancia del tráfico. El POLICY4 es un ejemplo de configuración del color de reconocimiento.

Ejemplo 2.6 Configuración de la vigilancia del tráfico

```
class-map match-all CLASS1
    match dscp ef
  class-map match-all CLASS2
    match dscp af11  af12   af13
  class-map match-all CLASS3
    match dscp af31
  class-map match-all CLASS4
    match dscp af32
  class-map match-all CLASS5
    match dscp af31  af32  af33
  !
  policy-map POLICY1
    class CLASS1
     police rate 1000000 burst 31250
    class CLASS2
     police rate 2000000 peak-rate 4000000
       conform-action transmit
       exceed-action transmit
  !
  policy-map POLICY2
    class class-default
     police rate percent 10 peak-rate percent 20
       conform-action set-mpls-exp-imposition-transmit 5
       conform-action set-qos-transmit 5
       exceed-action drop
  !
  policy-map POLICY3
    class class-default
     police rate 10000 cps atm-mbs 2500 peak-rate 20000 cps
       conform-action set-mpls-exp-imposition-transmit 1
       exceed-action set-mpls-exp-imposition-transmit 2
  !
```

```
policy-map POLICY4
  class CLASS5
   police rate 100000 peak-rate 200000
     conform-color CLASS3 exceed-color CLASS4
     conform-action set-dscp-transmit af31
     exceed-action set-dscp-transmit af32
     violate-action set-dscp-transmit af33
  class class-default
   police rate percent 10 peak-rate percent 20
!
```

2.3.4 Formación del tráfico

El comando *shape* configura la formación de tráfico y define una razón máxima de ancho de banda para una clase. Esto se aplica en el componente queuing del TMN y causa la cola del paquete cuando el tráfico que llega excede un perfil de tráfico. El formador hace cumplir el valor durante un intervalo de tiempo. Este intervalo se puede definir como la frecuencia con que el formador rellena la transmisión. Algunas formas de este comando permiten que usted controle este intervalo. En la Tabla 17 del Anexo 1 se muestran las opciones de configuración del comando *shape*. En la formación del tráfico se usan los comandos *shape average* y *shape peak* para hacer cumplir un valor máximo promedio y pico respectivamente. La clave *percent* habilita la definición de perfiles de tráfico relativos. Ambos, el *shape average* y el *shape speak* soportan esta palabra clave. [30]

El comando *shape adaptative* (Tabla 18 Anexo 1) ajusta la velocidad de formación en respuesta a la notificación de congestión. Este orden define una velocidad de formación reducida. La formación adaptable es útil para un ambiente de Frame Relay donde un dispositivo se informa sobre la congestión de la red por marcos que llegan con la bandera de notificación de congestión explícita atrasada (BECN, Backward Explicit Congestion Notification). Cuando un dispositivo recibe una notificación de congestión, el formador disminuye la velocidad de formación hasta que esta alcance el valor reducido configurado. Cuando llega la notificación de que la congestión cesa, el formador aumenta el valor de formación hasta el máximo original. El Ejemplo 2.7 muestra tres políticas de formación diferentes [35]. [30]

Ejemplo 2.7 Políticas para la Formación de Tráfico

```
class-map match-all CLASS1
    match cos  3  4
  class-map match-all CLASS2
    match cos  1  2
  !
  policy-map POLICY1
    class class-default
      shape average 1000000
  !
  policy-map POLICY2
    class class-default
      shape peak 1024000 4096 4096
      shape adaptive 1024000
  !
  policy-map POLICY3
    class CLASS1
      shape average percent 5
    class CLASS2
      shape average percent 10
  !
```

2.3.5 Mecanismos de control de la congestión

Los comandos *bandwidth*, *bandwidth remaining percent* y *priority* son los tres mecanismos principales que definen el queuing en el MQC. Estos comandos configuran el ancho de banda mínimo, el de exceso, y los atributos de prioridad del subcomponente dequeuing descrito anteriormente. La realización subyacente asigna las colas y configura los mecanismos de programación del paquete para satisfacer la asignación de ancho de banda y la asignación de prioridades de tráfico (Tabla 19 Anexo 1) que la política define. El comando *shape* complementa estos comandos permitiendo definir asignaciones de ancho de banda máximas. [30]

El comando *bandwidth* puede definir el ancho de banda mínimo que una cola recibe. La forma más simple del comando *bandwidth* especifica una garantía de ancho de banda mínima en términos absolutos. También se puede definir la garantía como un porcentaje del

valor de ancho de banda esencial usando la sintaxis *bandwidth percent*. El comando *bandwidth remaining percent* realiza la asignación de ancho de banda de exceso. El ancho de banda de exceso incluye el ancho de banda que no es parte de las garantías mínimas o el ancho de banda que otras clases no usan dentro de sus garantías mínimas en un momento determinado. [30]

El comando *priority* indica que una clase requiere una latencia baja por lo que el planificador debe servir el tráfico con estricta prioridad. El tráfico de prioridad no es parte del proceso de asignación que el planificador realiza para clases de no prioridad. Por lo tanto, la configuración *priority y bandwidth* es mutuamente exclusiva dentro de la misma clase. [30]

La configuración de una clase con prioridad influye en la configuración del ancho de banda de clases de no prioridad. En su forma más simple, el comando *priority* no usa ningún parámetro, y ningún límite superior de ancho de banda se aplica al tráfico de prioridad. Por lo tanto, no se puede asignar garantías de ancho de banda mínima a otras clases dentro de la misma política. Sin embargo, se puede asignar ancho de banda de exceso usando el *comando bandwidth remaining percent*. En cambio, se puede configurar explícitamente el controlador para que cumpla con un límite de ancho de banda para el tráfico de prioridad. En este caso, otras clases pueden recibir garantías de ancho de banda mínima iguales al ancho de banda esencial de la conexión menos la suma de los valores de todas las clases con prioridad. [30]

El comando *queue-limit* define el tamaño máximo de una cola en particular. Cuando una cola alcanza sus límites, el proceso de enqueuing deja caer los nuevos paquetes que llegan a la cola. Este comando usa paquetes como unidad de configuración por defecto. Se pueden definir límites de cola diferentes según las marca del paquete para poner en práctica la gota de la cola. Las Tablas 20 y 21 en el Anexo 1 muestran las diferentes formas que el comando *queue-limit* puede usar. [30]

El Ejemplo 2.8 muestra cuatro políticas de cola diferentes:

El POLICY1 clasifica el tráfico usando el campo EXP de MPLS. Esta política garantiza baja latencia para CLASS1 y limita el tráfico CLASS1 con 20,000,000 bps con un tamaño

de ráfagas de 25,000 bytes. CLASS2 y la clase por defecto, reciben una garantía de ancho de banda mínimo de 80,000 y 10,000 kbps respectivamente.

El POLICY2 especifica una política de cola que requiere un planificador de tres parámetros. La política define todas las razones y límites de cola en términos relativos. La CLASS2 recibe tanto una garantía mínima como de exceso de ancho de banda. El tráfico en la clase por defecto (class-default) obtiene sólo la asignación del ancho de banda de exceso.

El POLICY3 ilustra un tercer ejemplo que tiene dos clases de prioridad y asigna el ancho de banda a dos clases de no prioridad en forma de ancho de banda de exceso.

El POLICY4 muestra una política similar a la del POLICY3. En este caso, CLASS3 y CLASS4 tienen niveles de prioridad diferentes y ambos usan controladores explícitos.

Ejemplo 2.8 Políticas de cola

```
class-map match-all CLASS1
   match mpls experimental topmost 5
  class-map match-all CLASS2
   match mpls experimental topmost 3 4
  class-map match-all CLASS3
   match dscp ef
   match access-group 1
  class-map match-all CLASS4
   match dscp ef
   match access-group 2
  class-map match-all CLASS5
   match dscp af11 af12 af13
  !
  policy-map POLICY1
    class CLASS1
      priority
      police rate 20000000 burst 25000
    class CLASS2
      bandwidth 80000
      queue-limit 7500
    class class-default
      bandwidth 10000
      queue-limit 1250
```

```
!
policy-map POLICY2
  class CLASS1
    priority
    police rate percent 20
  class CLASS2
    bandwidth percent 50
    bandwidth remaining percent 25
    queue-limit 100 ms
  class class-default
    bandwidth remaining percent 75
    queue-limit 200 ms
!
policy-map POLICY3
  class CLASS3
    priority percent 5
  class CLASS4
    priority percent 20
  class CLASS5
    bandwidth remaining percent 50
  class class-default
    bandwidth remaining percent 50
!
policy-map POLICY4
  class CLASS3
    priority level 1
    police rate percent 5
  class CLASS4
    priority level 2
    police rate percent 20
  class CLASS5
    bandwidth remaining percent 50
  class class-default
    bandwidth remaining percent 50
!
```

2.3.6 Dirección de Cola Activa

La Dirección de Cola Activa (AQM, Active Queue Management) es configurada mediante el comando *random-detect* usando el algoritmo WRED (Weighted Random Early Detection). Esta función es parte del componente queuing. La configuración de WRED define dos elementos principales: ponderación del campo y los límites. Se tiene que especificar la ponderación del campo primero (por ejemplo, precedencia de IP, DSCP, clase de descarte), para luego especificar los umbrales para una marca en particular del campo ponderado. Las Tablas 22 y 23 en el Anexo 1 muestran el uso de estos elementos en el WRED. [30]

Mediante el comando *random-detect ecn* se puede configurar el WRED para realizar la notificación de congestión explícita (ECN, Explicit Congestion Notification) para el tráfico IP. El ECN afecta la operación del WRED entre el umbral mínimo y el umbral máximo. El WRED comprueba el campo de ECN antes de decidir la acción apropiada a tomar con un paquete cuando este ha seleccionado aquel paquete para la caída. Si el campo de ECN indica que la cola del paquete es posible para el ECN, WRED marca la indicación de congestión en el campo de ECN, y el paquete entra en su cola respectiva. Si el campo de ECN ya contiene una indicación de congestión, el WRED también permite que el paquete entre en su cola. Por otra parte, si el campo de ECN indica que la cola del paquete no es posible, WRED deja caer el paquete como esto normalmente ocurre [25]. [30]

En el Ejemplo 2.9 incluye tres políticas que realizan la dirección de cola activa con WRED. La primera política, el POLICY1, tanto para CLASS2 como para la clase por defecto hacen el uso de WRED en base a precedencia y define explícitamente los umbrales en términos de números de paquetes. El POLICY2 habilita WRED basado en la clase de descarte de cada paquete. La política especifica umbrales explícitos en milisegundos los valores de DSCP, cero, uno, y dos. El POLICY3 habilita WRED basado en la clase de descarte de cada paquete. La política especifica umbrales explícitos en milisegundos para valores de DSCP CS1, CS2, y por defecto. El POLICY4 habilita la notificación de congestión para el tráfico de IP usando WRED en base a precedencia y el ECN. Esta política no define ningún umbral explícitamente. Por lo tanto, WRED usa valores de umbral por defecto para los ocho valores diferentes de precedencia de IP. [35]

Ejemplo 2.9 Control de la cola mediante el uso de WRED

```
class-map match-all CLASS1
    match mpls experimental topmost 5
  class-map match-all CLASS2
    match mpls experimental topmost 3 4
  class-map match-all CLASS3
    match dscp cs1 cs2
  !
  policy-map POLICY1
    class CLASS1
      priority
      police rate 2000000
    class CLASS2
      bandwidth 10000
      random-detect
      random-detect precedence 3 2000 4000 1
      random-detect precedence 4 4000 6000 1
    class class-default
      random-detect
      random-detect precedence 0 4000 6000 1
  !

  policy-map POLICY2
    class class-default
      random-detect discard-class-based
      random-detect discard-class 0 75 ms 150 ms 1
      random-detect discard-class 1 25 ms 150 ms 1
      random-detect discard-class 2 75 ms 150 ms 1
  !
  policy-map POLICY3
    class CLASS3
      bandwidth percent 40
      random-detect dscp-based
      random-detect dscp cs1 25 ms 75 ms
      random-detect dscp cs2 50 ms 100 ms
    class class-default
      bandwidth percent 40
      bandwidth remaining percent 60
      random-detect dscp-based
```

```
      random-detect dscp default 25 ms 100 ms
  !
policy-map POLICY4
   class class-default
    random-detect
    random-detect ecn
  !
```

2.3.7 Fragmentación e Intercalado del Tráfico

Cisco IOS soporta el LFI con el multienlace MLP (Multilink PPP) y la encapsulación Frame Relay. La configuración de la fragmentación e intercalando (LFI) requiere la definición de un tamaño de fragmento. La configuración es específica al tipo de fragmentación que se está utilizando (MLP o Frame Relay FRF.12). El comando **ppp** *multilink fragment delay* define el tamaño del fragmento (en milisegundos o bytes) para MLP. El comando ***ppp multilink interleave*** habilita el LFI. En caso de los enlaces Frame Relay, el comando *frame-relay fragment* define el tamaño del fragmento (en bytes). El LFI no requiere la configuración explícita en este caso. Estos comandos requieren el uso simultáneo de una política con una clase de prioridad en la interfase o PVC donde usted ha habilitado la fragmentación. La Tabla 24 del Anexo 1 resume los comandos del LFI. [25, 30]

El Ejemplo 2.9 muestra la configuración para el LFI usando MLP y Frame Relay FRF.12. El POLICY1 presenta una política de cola simple con CLASS1 sirviendo el tráfico de prioridad y las demás clases sirviendo el resto del tráfico. La interfase *Serial1/0:0* usa MLP para objetivos LFI. La interfase es parte del grupo *multilink group 1* asociado con la interfase *Multilink1*. Esta interfase habilita el intercalado, y define el tamaño de fragmento como 480 bytes. Del mismo modo, la interfase Serial1/0:1 pone en práctica LFI para Frame Relay PVC y define un tamaño de fragmento de 480.

Ejemplo 2.9 Políticas con LFI

```
class-map match-all CLASS1
    match dscp ef
  !
  policy-map POLICY1
    class CLASS1
      priority
      police rate percent 25 burst 10 ms
  !
  interface Multilink1
   ip address 192.168.2.1 255.255.255.252
   ppp multilink
   ppp multilink interleave
   ppp multilink group 1
   ppp multilink fragment delay size 480
   service-policy output POLICY1
  !
  interface Serial1/0:0
   no ip address
   encapsulation ppp
   ppp multilink
   ppp multilink group 1
  !
  interface Serial1/0:1
   ip address 192.168.2.5 255.255.255.252
   encapsulation frame-relay
   frame-relay interface-dlci 16
   frame-relay fragment 480 end-to-end
   service-policy output POLICY1
  !
```

2.3.8 Compresión de la cabecera

El comando *compression* habilita la compresión de la cabecera en una política. La compresión del encabezamiento es una acción del componente prequeuing en el modelo de funcionamiento de QoS en CISCO. Es posible configurar la compresión de la cabecera RTP y TCP para mayor eficiencia del ancho de banda. La política realiza la compresión en

ambos protocolos si usted no configura explícitamente uno de los dos. Esta compresión de encabezamiento es una tecnología de punto a punto (point-to-point) y requiere que ambos extremos participen en la operación. La Tabla 6 muestra la sintaxis completa de este comando. [25, 30]

El Ejemplo 2.10 muestra una política de compresión de cabecera. La primera clase (CLASS1) realiza la compresión de la cabecera RTP mientras que la otra clase (CLASS2) realiza ambas. Los paquetes que pertenezcan a la clase por defecto (class-default) no están sujetos a ningún tipo de compresión.

Ejemplo 2.10 Políticas de Compresión de Cabecera

```
class-map match-all CLASS1
   match dscp ef
  class-map match-all CLASS2
   match dscp cs4
  !
  policy-map POLICY1
   class CLASS1
     priority percent 25
     compress header ip rtp
   class CLASS2
     bandwidth percent 50
     compress header ip
  !
```

2.4 Configuraciones jerárquicas

Las configuraciones jerárquicas permiten ampliar las capacidades del MQC. Estas configuraciones habilitan la referencia a una política dentro de otra política. Del mismo modo, un mapa de clase puede referirse a otros mapas de clase existentes. Las políticas jerárquicas hacen posible realizar diferentes acciones en subclases del tráfico en niveles diferentes. Además, un acercamiento jerárquico habilita la creación de módulos de configuración que usted puede reutilizar repetidamente en otras políticas. [25, 30]

2.4.1 Clasificación jerárquica

Las configuraciones jerárquicas de mapa de clase hacen posible elaborar criterios de clasificación utilizando operaciones lógicas entre criterios de correspondencia. El comando *match class-map* mostrado en la Tabla 7 del Anexo 1 habilita este tipo de configuraciones jerárquicas. Es posible definir operaciones lógicas complejas entre criterios de correspondencia en la combinación con los comandos *match-any* y *match-all*. [25, 30]

El CLASS1 en el Ejemplo 2.11 tiene una configuración jerárquica que se refiere a CLASS2. En este ejemplo, un paquete IP pertenecerá a CLASS1 si este tiene un valor EF del DSCP o si este satisface los criterios de clasificación de CLASS2. Un paquete pertenece a CLASS2 si este tiene un valor de DSCP de CS5 y satisface la lista de acceso ACL1.

Ejemplo 2.11 Clasificación Jerárquica

```
class-map match-any CLASS1
   match dscp ef
   match class-map CLASS2
  class-map match-all CLASS2
   match dscp cs5
   match access-group name ACL1
  !
```

2.4.2 Políticas jerárquicas

El MQC soporta la configuración de políticas jerárquicas donde una clase dentro de una política se hace el punto de acceso para otra política. Muchas combinaciones de acciones son posibles dentro de una política jerárquica (por ejemplo, vigilancia jerárquica y formación de una cola jerárquica).

Las implementaciones de políticas jerárquicas normalmente limitan la jerarquía con dos o tres niveles. Una política que incluye otra política recibe el nombre de una política de padre. La política que la política paternal incluye recibe el nombre de una política de hijo. En una jerarquía de tres niveles, una política puede ser un abuelo o un nieto de otra política.

La política de hijo se aplica usando el comando *service-policy*. La política de hijo automáticamente hereda la política de dirección su padre.

Las políticas jerárquicas usan un solo paso de clasificación. Por lo tanto, la clasificación de un paquete permanece sin alterar aun si una acción en una política de nieto o hijo lo remarca. La nueva marca no causará la nueva clasificación del paquete. Sin embargo, la nueva marca afectará la operación del WRED o los límites de cola ponderados. [25, 30]

Las acciones tienen un orden particular ejecución en una configuración jerárquica. Después de la clasificación, las políticas jerárquicas ejecutan acciones en el nivel decreciente. Por ejemplo, en una política de tres niveles, la política de nieto ejecuta sus acciones en el paquete antes de la política de padre. Del mismo modo, éste ejecutará sus acciones antes de la política de abuelo. Este ordenamiento se aplica a todas las acciones a excepción del comando *set*. Por ejemplo, en una política que usa este comando en todos los niveles, la marca final del paquete corresponderá al valor que el comando *set* indica en el instante del nieto.

En el Ejemplo 2.12, el POLICY1 define una política jerárquica de dos niveles. El POLICY1 forma todo el tráfico a 10,000,000 bps e invoca al POLICY2 como una política de hijo. El POLICY2 define una política de cola que proporciona baja latencia al tráfico con un valor EF de DSCP y habilita el WRED en el tráfico restante.

Ejemplo 2.12 Política jerárquica de dos niveles

```
class-map match-all CLASS1
    match dscp ef
  !
  policy-map POLICY1
    class class-default
     shape average 10000000 40000 40000
     service-policy POLICY2
  !
  policy-map POLICY2
    class CLASS1
      priority percent 20
    class class-default
      random-detect dscp-based
```

```
    !
```

El Ejemplo 2.13 amplía la jerarquía un nivel para producir una política jerárquica de tres niveles. En este ejemplo, el POLICY2 invoca el POLICY3 como una política de hijo. El POLICY3 marca la clase CLASS2 usando un controlador y todo el otro tráfico con un comando *set*. El POLICY2 usa la nueva marca de paquete realizando WRED en la clase por defecto.

Ejemplo 2.13 Política jerárquica de tres niveles

```
class-map match-all CLASS1
  match dscp ef
class-map match-all CLASS2
  match access-group name ACL1
!
policy-map POLICY1
 class class-default
  shape average 10000000 40000 40000
  service-policy POLICY2
!
policy-map POLICY2
  class CLASS1
   priority percent 20
  class class-default
   random-detect dscp-based
   service-policy POLICY3
!
policy-map POLICY3
  class CLASS2
   police rate percent 50
     conform-action set-dscp-transmit af21
     exceed-action set-dscp-transmit af22
  class class-default
   set dscp af21
   !
```

2.5 Razones a base de porcentaje

El punto de acceso de una política determina la razón de datos actual que los comandos *bandwidth shape* y *police* usan cuando se configuran las razones a base de porcentaje. El comando *bandwidth percent* define una garantía mínima de amplitud de banda con relación a la garantía mínima de amplitud de banda del punto de acceso de la política en la política paternal. Los comandos *shape* y *police* usan la razón máxima del punto de acceso como una referencia. La presencia de los comandos *shape* y *police* en la política paternal define el rate máximo. En ausencia de esos comandos en la política paternal, la política paternal hereda el máximo de su padre (política de abuelo) y finalmente del interfaz que sirve como el punto de acceso para la política jerárquica [25, 30].

El comando *priority percent* usa la misma lógica del *bandwidth percent* para calcular la razón del controlador condicional.

Los interfaces generalmente tienen una definición de amplitud de banda implícita que las políticas pueden usar como una referencia. En algunos casos, el interfaz no tendrá una cantidad de amplitud de banda asociada, y se puede tener que especificar la razón de datos del interfaz. Esta situación puede ser común en particular en subinterfaces lógicas (por ejemplo, Ethernet, Frame Relay, or ATM). El comando *bandwidth qos-reference* especifica la cantidad de amplitud de banda que las políticas deberían usar como una referencia en un interfaz. Esta orden se aplica directamente bajo la configuración del interfaz, y cualquier política que se ate al interfaz usará automáticamente aquella referencia de amplitud de banda[30]. La Tabla 26 del Anexo 1 muestra la sintaxis completa del comando *bandwidth qos-reference.* [25, 30]

2.6 Unidades de los parámetros

Las Tablas 27 y 28 en el Anexo 1 resumen las unidades por defecto de los parámetros usados en los comandos del MQC. Ambas incluyen una sintaxis simplificada de los comandos. La Tabla 27 del Anexo 1 incluye comandos que tienen rates (razón de datos) o bursts (ráfagas) como parámetros. La Tabla 28 del Anexo 1 incluye comandos con un tamaño de cola como parámetros. [29]

Algunas redes soportan la configuración explícita de las unidades de los parámetros en su implementación del MQC. La definición flexible de unidades facilita la operación de red y contribuye a menos errores de configuración. Sus ventajas se hacen más obvias como el aumento de velocidades de enlace y las unidades por defecto para un parámetro se hacen menos adecuadas. La Tabla 29 del Anexo 1 muestra las diferentes claves para las unidades de rate. Estas palabras claves demuestran utilidades en la configuración de policers y shapers o cuando se realiza la asignación de amplitud de banda para la dirección de la congestión. La Tabla 30 del Anexo 1 muestra las palabras claves para las unidades de memoria. Estas palabras claves facilitan la configuración de tamaños de ráfagas para policers y shapers. Estas también facilitan la definición del tamaño máximo para una cola o umbrales para la dirección de cola activa. Finalmente, la Tabla 31 del Anexo 1 incluye unidades de tiempo que ayudan a definir tamaños de ráfagas para policers y shapers. [29, 30]

El Ejemplo 2.14 muestra una política que configura explícitamente las unidades de los parámetros. La clase CLASS1 usa un controlador con un rate en Mbps y una ráfaga en kilo-bytes. CLASS2 tiene una garantía mínima de amplitud de banda en Mbps. Tanto CLASS2 como la clase por defecto (class-default) tienen un tamaño de cola máximo en paquetes.

Ejemplo 2.14 Política con Unidades de Parámetro Explícitas

```
class-map match-all CLASS1
   match mpls experimental topmost 5
  class-map match-all CLASS2
   match mpls experimental topmost 3 4
  !
  policy-map POLICY1
   class CLASS1
    priority
    police rate 1 mbps burst 3 kbytes
  class CLASS2
    bandwidth 10 mbps
    queue-limit 1000 packets
   class class-default
    queue-limit 8700 packets
   !
```

2.7 Procesamiento local del tráfico

El procesamiento del tráfico local representa un caso especial en la configuración de QoS. Este tráfico representa sobre todo paquetes de control y dirección plana que un nodo recibe y envía. Los nodos identifican estos paquetes con una bandera de prioridad. Las políticas de interfaz de QoS no afectan estos paquetes. Por ejemplo, el controlador o AQM no dejan caer paquetes que tienen una bandera de prioridad. Las políticas de interfaz generalmente clasifican y tratan todo el otro tráfico local.

La marca del tráfico IP local también representa un caso especial en la configuración de QoS. El nodo automáticamente marca una parte de aquel tráfico. En algunos casos, es posible preconfigurar la marca que se quiere fuera del MQC utilizando protocolo o comandos de aplicación fuera del MQC. La Tabla 3 del Anexo 2 muestra la lista de fuentes de tráfico IP locales que no usan una marca de mejor esfuerzo por defecto. [30]

2.8 Conclusiones del capítulo

En este capítulo después de una explicación paso a paso del funcionamiento de QoS en los productos Cisco se puede concluir que el TMN y el MQC brindan grandes facilidades para la configuración de la QoS. Se destacan las ventajas que tiene establecer el MQC proporcionando un modelo de configuración a base de plantilla, independientemente de las diferentes plataformas de Cisco que se utilice. [30]

CAPÍTULO 3. Configuración de MPLS DiffServ en un backbone IP/MPLS.

En el presente capítulo se implementa MPLS DiffServ para una red IP/MPLS mediante el software Cisco IOS [36] y Cisco IOS XR [31]. Para la configuración de QoS se toma un modelo IP/MPLS de referencia y se examinan varias alternativas de diseño que se pueden utilizar para implementar MPLS DiffServ [25].

3.1 Modelo de red de referencia

La Figura 3.1 muestra el modelo de red que se ha tomado como referencia para diseñar el modelo de Servicios Diferenciados. Este backbone IP/MPLS presenta seis nodos de borde (PE, provider edge) y tres nodos interiores (P, provider), los nombres P y PE indican el papel que realizan estos en la red. Para todas las direcciones IP del backbone se ha utilizado la subred 172.16.0.0/24.

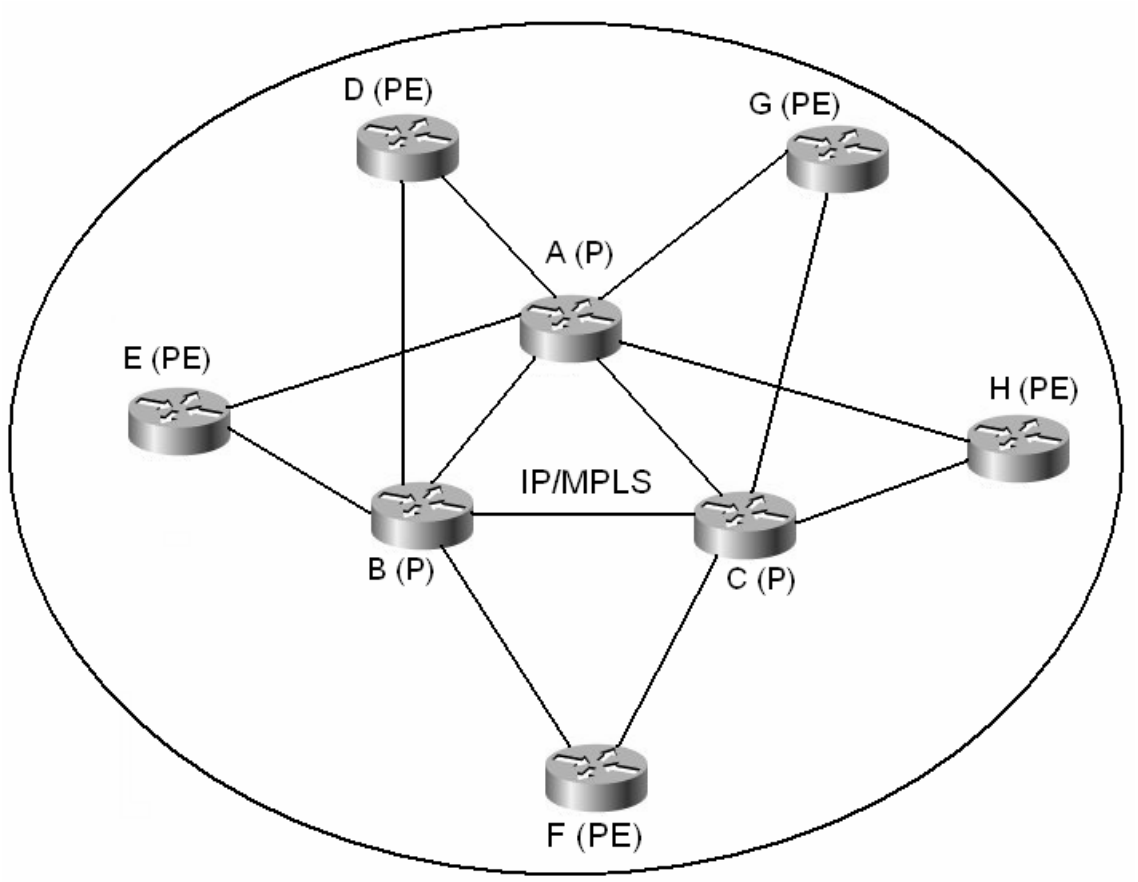

Figura 3.1 Red de referencia

A continuación se muestra en las Tablas 3.1 a 3.8 la lista de direcciones para cada nodo.

Tabla 3.1 Interfaz de información para el nodo E [EP]

Interfaz	**Dirección IP**	**Vecino**
Loopback0	172.16.255.1/32	E [PE]
POS0/1/0	172.16.0.0/31	B [P]
POS1/0/0	172.16.0.2/31	A [P]

Tabla 3.2 Interfaz de información para el nodo H [EP]

Interfaz	**Dirección IP**	**Vecino**
Loopback0	172.16.255.2/32	H [PE]
POS2/0	172.16.4.0/31	C [P]
POS2/1	172.16.4.2/31	A [P]

Tabla 3.3 Interfaz de información para el nodo F [EP]

Interfaz	**Dirección IP**	**Vecino**
Loopback0	172.16.255.3/32	F [PE]
POS1/0	172.16.8.0/31	B [P]
POS1/1	172.16.8.2/31	C [P]

Tabla 3.4 Interfaz de información para el nodo D [EP]

Interfaz	**Dirección IP**	**Vecino**
Loopback0	172.16.255.4/32	D [PE]
POS2/1	172.16.0.6/31	A [P]

Interfaz	Dirección IP	Vecino
POS2/2	172.16.0.4/31	B [P]

Tabla 3.5 Interfaz de información para el nodo G [EP]

Interfaz	Dirección IP	Vecino
Loopback0	172.16.255.5/32	G [PE]
POS2/1	172.16.4.6/31	A [P]
POS2/2	172.16.4.4/31	C [P]

Tabla 3.6 Interfaz de información para el nodo B [EP]

Loopback0	172.16.255.129/32	B [P]
POS0/3/0/0	172.16.0.1/31	E [PE]
POS0/3/0/1	172.16.192.0/31	C [P]
POS0/3/0/2	172.16.192.2/31	A [P]
POS0/3/0/3	172.16.0.5/31	D [PE]
POS0/3/0/4	172.16.8.1/31	F [PE]

Tabla 3.7 Interfaz de información para el nodo C [EP]

Interfaz	**Dirección IP**	**Vecino**
Loopback0	172.16.255.130/32	C [P]
POS0/0/0	172.16.192.1/31	B [P]
POS0/1/0	172.16.4.5/31	G [PE]
POS1/0/0	172.16.8.3/31	F [PE]
POS1/1/0	172.16.192.4/31	A [P]
POS2/0/0	172.16.4.1/31	H [PE]

Tabla 3.8 Interfaz de información para el nodo A [EP]

Interfaz	**Dirección IP**	**Vecino**
Loopback0	172.16.255.131/32	A [P]
POS0/0/0	172.16.192.3/31	B [P]
POS0/1/0	172.16.4.7/31	G [PE]
POS0/1/1	172.16.4.3/31	H [PE]
POS1/0/0	172.16.192.5/31	C [P]

Interfaz	Dirección IP	Vecino
POS1/1/0	172.16.0.7/31	D [PE]
POS2/0/0	172.16.0.3/31	E [PE]

3.2 Alternativas de diseño para implementar DiffServ en IP/MPLS

Mediante la diferenciación del tráfico en el backbone IP/MPLS, el modelo de Servicios Diferenciados realiza la planificación de la capacidad por clases y usa objetivos de utilización diferentes para las diferentes clases según sus exigencias de rendimiento[24]. En un camino, el despliegue de DiffServ crea redes virtuales múltiples, donde usted realiza la planificación de la capacidad independientemente como se muestra en la Figura 3.2:

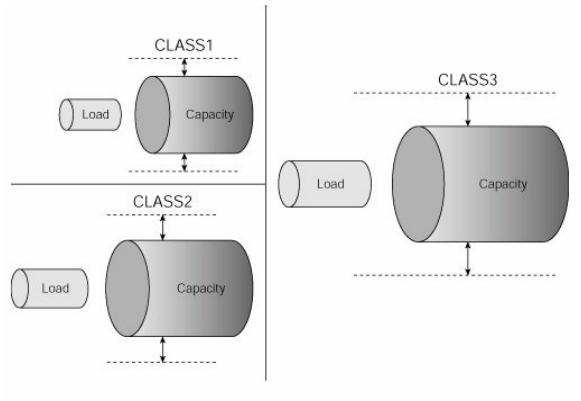

Figura 3.2 Clases diferenciadas

Para poner en práctica el modelo DiffServ en una red MPLS es necesario definir las diferentes clases y sus marcas de paquetes respectivas. El campo EXP de MPLS proporciona los valores suficientes para satisfacer las exigencias de los despliegues más comunes. La Tabla 3.9 muestra cuatro diseños con usos diferentes del campo EXP. En general, el control de tráfico de la red debería hacer el uso de los valores EXP seis y siete,

así como el valor cinco para el tráfico de EF y el valor cero para el tráfico por defecto o de mejor esfuerzo. Por ultimo, los valores del uno al cuatro para definir otras clases, incluso AF. Éstos no son obviamente los únicos diseños posibles, pero siguen las mejores prácticas comunes. [25]

Tabla 3.9 Alternativas para la asignación de los valores EXP de MPLS

MPLS EXP	Diseño 1 (EF y DF)	Diseño 2 (EF, AF, y DF)	Diseño 3 (EF, AF1, AF2, y DF)	Diseño 4 (EF, CS4, CS3, AF, y DF)
7	Reservado (control de la red)	Reservado (control de la red)	Reservado (control de la red)	Reservado (control de la red)
6	Reservado (control de la red)	Reservado (control de la red)	Reservado (control de la red)	Reservado (control de la red)
5	CLASS1	CLASS1	CLASS1	CLASS1
4	No es usado	No es usado	CLASS2; baja probabilidad de gota	CLASS2
3	No es usado	No es usado	CLASS2; alta probabilidad de gota	CLASS3
2	No es usado	CLASS2; baja probabilidad de gota	CLASS3; baja probabilidad de gota	CLASS4; baja probabilidad de gota
1	No es usado	CLASS2; alta probabilidad de gota	CLASS3; alta probabilidad de gota	CLASS4; alta probabilidad de gota
0	CLASS2	CLASS3	CLASS4	CLASS5

La mayor parte de las utilizaciones DiffServ usan relativamente un número pequeño de colas (típicamente entre dos y cuatro colas) en el backbone [24]. Esta simplificación facilita la operación y la dirección de la red, pero requiere que el diseño apropiado garantice todas las exigencias de rendimiento requeridas. La Tabla 3.10 muestra tres diseños comunes para la realización de DiffServ. Todos ellos usan una cola dedicada al tráfico de tiempo real y una cola separada que no proporciona ninguna garantía de funcionamiento. Los diseños 2 y 3 hacen el uso de colas adicionales que tienen pérdidas bajas especiales. Se puede trazar un mapa de clases con exigencias de interpretación similares a la misma cola. En general, el número de colas es igual o inferior que el número de clases. La mayor parte de despliegues se parecen a uno de estos diseños. [5, 25, 30, 31, 35]

Tabla 3.10 Alternativas para la implementación DiffServ

Garantías de funcionamiento	Diseño 1	Diseño 2	Diseño 3
Latencia baja, jitter bajo, pérdidas bajas (sin AQM)	Queue(cola)1	Queue1	Queue1
Pérdidas bajas (sin AQM)			Queue2
Pérdidas bajas (AQM)		Queue2	Queue3
Sin garantías (AQM)	Queue2	Queue3	Queue4

3.3 Configuración de MPLS DiffServ mediante Cisco IOS y Cisco IOS XR

A continuación se presentan dos ejemplos de configuración:

El primer ejemplo, se implementa en el nodo G [EP] mediante el software Cisco IOS [36]**,** utilizando un planificador de dos parámetros y cuatro colas. Los paquetes con un valor cinco de del campo EXP de MPLS pertenecen a la clase uno (CLASS1). Esta política proporciona tratamiento EF a la clase1. La clase dos (CLASS2) incluye los paquetes IP con un valor DSCP de CS6 y los paquetes MPLS con valores cuatro y seis del campo EXP.

Esta atiende el control del tráfico y la interacción de los datos sin la optimización de TCP y con demoras y pérdidas mas bajas. Los paquetes con valores uno y dos del campo EXP de MPLS conforman la clase tres (CLASS3), que proporciona el tratamiento AF para el tráfico de datos de alto rendimiento y usa WRED para AQM para proporcionar la caída diferenciada de paquetes y la optimización TCP. Por último, la clase por defecto sirve el resto del tráfico (valores cero de EXP) solamente con la optimización TCP. Para la política de este ejemplo se han usado los parámetros apropiados para un interfaz OC/ STM-1. La clase uno recibe el servicio de latencia baja, y el controlador limita este al 50% del ancho de banda total (77.5 Mbps) con un burst de 20 ms (193.750 bytes). La clase dos tiene una garantía mínima de ancho de banda de 20% y un tamaño máximo de cola de 757 paquetes (50 ms para paquetes de 256 bytes). La clase tres tiene la misma garantía de ancho de banda pero usa WRED para implementar el comportamiento diferenciado de gota. Los paquetes con valor uno del campo EXP de MPLS presentan una alta probabilidad de gota y experimentan umbrales más agresivos (entre 151 y 500 paquetes, o 10 ms y 33 ms respectivamente). Un valor de dos indica una probabilidad de gota menor y umbrales menos agresivos (entre 100 y 1514 paquetes, o 33 ms y 100 ms respectivamente). El tamaño máximo de cola para la clase tres es de 2271 paquetes (150 ms). Por último, la clase por defecto recibe una garantía de ancho de banda del 10%, usa WRED para la optimización TCP (entre 378 y 1135 paquetes, o 50 ms y 150 ms respectivamente), y tiene un tamaño máximo de cola de 1514 paquetes (200 ms). [31, 36]

Ejemplo 3.1 Configuración MPLS DiffServ en el nodo G [PE] usando Cisco IOS

```
hostname G[PE]
 !
 mpls label protocol ldp
 !
 class-map match-all CLASS1
   match mpls experimental topmost 5
 class-map match-any CLASS2
   match mpls experimental topmost 4 6
   match dscp cs6
   class-map match-all CLASS3
```

```
 match mpls experimental topmost 1 2
!
policy-map OUT-POLICY
  class CLASS1
    priority
    police rate 50 burst 20 ms
  class CLASS2
    bandwidth percent 20
    queue-limit 757
  class CLASS3
    bandwidth percent 20
    random-detect
    random-detect precedence 1 151 500 1
    random-detect precedence 2 86 1514 1
    queue-limit 387
  class class-default
    random-detect
    random-detect precedence 0 378 1135 1
    queue-limit 1514
    bandwidth percent 10
!
interface Loopback0
 ip address 172.16.255.5 255.255.255.255
!
interface POS2/1
 description CONNECTS TO A[P]
 ip address 172.16.4.6 255.255.255.254
 encapsulation ppp
 mpls ip
 service-policy output OUT-POLICY
!
interface POS2/2
 description CONNECTS TO C[P]
 ip address 172.16.4.4 255.255.255.254
 encapsulation ppp
 mpls ip
 service-policy output OUT-POLICY
!
router ospf 100
 log-adjacency-changes
```

```
    passive-interface Loopback0
    network 172.16.0.0 0.0.255.255 area 0
   !
```

El segundo ejemplo, se implementa en el nodo B [P] mediante el sotfware Cisco IOS XR. Este ejemplo muestra una realización de MPLS DiffServ con cuatro colas y un planificador de tres parámetros usando Cisco IOS XR [35]. La política de QoS usa el mismo número de clases con las mismas características de rendimiento. La diferencia principal es el uso de un planificador de tres parámetros. En este caso, la clase dos, clase tres, y la clase por defecto tienen garantías de ancho de banda mínimo y de exceso. La clase dos tiene más acceso al ancho de banda de exceso, seguido por la clase tres y luego la clase por defecto. Esta política mejora ligeramente las características de rendimiento para la clase dos y la clase tres cuando se compara con la política anterior. Se usan los mismos parámetros para los tamaños máximos de cola y umbral WRED que se usaron en el ejemplo anterior. [31, 36]

Ejemplo 3.2 Configuración MPLS DiffServ en el nodo B [P] usando Cisco IOS XR

```
hostname B[P]
   router-id Loopback0
   class-map match-any CLASS1
    match mpls experimental topmost 5
   !
   class-map match-any CLASS2
    match mpls experimental topmost 4 6
    match dscp ipv4 cs6
   !
   class-map match-any CLASS3
    match mpls experimental topmost 1 2
   !
   policy-map OUT-POLICY
    class CLASS1
     police rate percent 50 burst 20 ms
     priority
```

```
 !
 class CLASS2
  queue-limit 757
  bandwidth percent 20
  bandwidth remaining percent 60
 !
 class CLASS3
  queue-limit 2271
  random-detect exp 1 151 500
  random-detect exp 2 500 1514
  bandwidth percent 20
  bandwidth remaining percent 30
 !
 class class-default
  queue-limit 1514
  random-detect exp 0 378 1135
  bandwidth percent 10
  bandwidth remaining percent 10
 !
!
interface Loopback0
 ipv4 address 172.16.255.129 255.255.255.255
!
interface POS0/3/0/0
 description CONNECTS TO E[PE]
 service-policy output OUT-POLICY
 ipv4 address 172.16.0.1 255.255.255.254
 encapsulation ppp
!
interface POS0/3/0/1
 description CONNECTS TO C[P]
 service-policy output OUT-POLICY
 ipv4 address 172.16.192.0 255.255.255.254
 encapsulation ppp
!
interface POS0/3/0/2
 description CONNECTS TO A[P]
 service-policy output OUT-POLICY
 ipv4 address 172.16.192.2 255.255.255.254
 encapsulation ppp
```

```
  !
  interface POS0/3/0/3
   description CONNECTS TO D[PE]
   service-policy output OUT-POLICY
   ipv4 address 172.16.0.5 255.255.255.254
   encapsulation ppp
  !
  interface POS0/3/0/4
   description CONNECTS TO F[PE]
   service-policy output OUT-POLICY
   ipv4 address 172.16.8.1 255.255.255.254
   encapsulation ppp
  !
  router ospf DEFAULT
   area 0
  interface Loopback0
   passive
  !
  interface POS0/3/0/0
  !
  interface POS0/3/0/1
  !
  interface POS0/3/0/2
  !
  interface POS0/3/0/3
  !
  interface POS0/3/0/4
  !
 !
!
mpls ldp
 interface POS0/3/0/0
 !
 interface POS0/3/0/1
 !
 interface POS0/3/0/2
 !
 interface POS0/3/0/3
 !
 interface POS0/3/0/4
```

```
 !
!
```

3.4 Conclusiones del capítulo

Después del análisis de las distintas alternativas para implementar servicios diferenciados en un backbone IP/MPLS y la configuración de MPLS DiffServ, se puede llegar a la conclusión que tanto Cisco IOS como Cisco IOS XR permiten al administrador múltiples opciones de diseño que varían en previsión y complejidad. Todas estas opciones de diseño tienen el potencial para soportar un grupo dado de exigencias de funcionamiento del backbone.

Conclusiones

MPLS es un protocolo que soporta cualquier tipo de tráfico en una red IP sin depender de los protocolos de enrutamiento, capa de transporte y esquema de direccionamiento. El proceso de conmutación es realizado a nivel de hardware, obteniendo mejores prestaciones para aplicaciones de tiempo real. En correspondencia con los objetivos trazados en este trabajo se arribaron a las siguientes conclusiones:

1. La Calidad de Servicio es una herramienta que da al administrador un mayor control sobre su red, lo que significa menores costos y mayor satisfacción del cliente o usuario final.

2. El MPLS DiffServ sirve como herramienta clave para manipular la utilización de ancho de banda en todos los nodos de la red, aumentando las ventajas sobre los servicios diferenciados ya existentes.

3. Cisco se ocupa de desarrollar su propio software de gestión y configuración. A través de IOS e IOS XR se configura la QoS en los productos de Cisco mediante el MQC (Modular QoS Command-Line Interface), que sirve de intérprete entre el usuario y el equipo.

4. La configuración de MPLS DiffServ en Cisco ofrece múltiples opciones que varían en previsión y complejidad de acuerdo a las necesidades de funcionamiento y rendimiento de la red en particular.

Recomendaciones

Una vez concluido el trabajo se proponen las siguientes recomendaciones:

1. Continuar con el estudio de la Calidad de Servicio como herramienta para optimizar el uso de los recursos de la red.

2. Continuar el estudio de las diferentes realizaciones de Calidad de Servicio para IP/MPLS en Cisco debido a sus constantes transformaciones.

3. Implementar Ingeniería de Tráfico en MPLS DiffServ para complementar la optimización del ancho de banda de la red.

GLOSARIO

ATM: Asynchronous Transfer Mode, Modo de Transferencia Asincrónico.

BGP: Border Gateway Protocol, Protocolo de Borde.

CoS: Class of Service, Clases de Servicio.

CR-LDP: Constraint-Based Routing Label Distribution Protocol.

DiffServ: Differenciated Services, Servicios Diferenciados.

DLCI: Data Link Connection Identifier, Identificador de la Conexión del Enlace de Datos.

FEC: Class Forward Equivalence, Clase de Equivalencia de Direccionamiento.

FIB: Forward Information Base, Base de Información de Envío.

FLIB: Forward Label Information Base, Base de Información de Envío de Etiqueta. **FR:** Frame Relay, Retransmisión de Tramas.

FTP: File Transfer Protocol, Protocolo para la Transferencia de Ficheros

IETF: Internet Engineering Task Force, Fuerza de Trabajo de Ingenieria de Internet.

IP: Internet Protocol, Protocolo de Internet.

IS-IS: Intermedia System to Intermedia System, Sistema Intermedio-Sistema Intermedio.

LAN: Local Area Network, Red de Área Local.

LDP: Label Distribution Protocol, Protocolo de Distribución de Etiqueta.

LER: Label Edge Router, Enrutador de Borde de Etiqueta.

LIB: Label Information Base, Base de Información de Etiqueta.

LP: Labeled Packets, Paquetes Etiquetados.

LSP: Label Switched Path, Camino Conmutado de Etiqueta.

LSR: Label Switching Router, Enrutador de Intercambio de Etiqueta.

MAN: Metropolitan Area Network, Red de Área Metropolitana.

MPLS: Multi-Protocol Label Switching, Multiprotocolo de Conmutación basado en Etiquetas.

Multicast: Multidifusión.

OSPF: Open Shortest Path First, Primer Camino Abierto más Corto.

OPNET: Optimun Network Performance, Herramienta de Simulación de redes Óptimas.

Paquete *hello:* Paquete de multidifusión.

PC: Personal Computer, Computadora Personal.

PNNI: Private Network-Network Interface, Interface Red-Red Privada.

PPP: Point to Point Protocol, Protocolo Punto a Punto.

PVC: Permanent Virtual Circuit, Circuito Virtual Permanente.

QoS: Quality of Service, Calidad de Servicio.

RIP: Routing Information, Protocolo de Información de Rutas.

RSVP: Resource Reservation Protocol, Protocolo de Reserva de Recursos.

TCP: Transmission Control Protocol, Protocolo de Control de Transmisión.

ToS: Type of Service, Tipo de Servicio.

TTL: Time To Live, Tiempo de Vida.

UDP: User Datagram Protocol, Protocolo de Datagrama de Usuario.

Unicast: Unidifusión.

VPI/VCI: Virtual Path Identifier/ Virtual Canal Identifier, Identificador de Camino Virtual/ Identificador de Canal Virtual

VPN: Network Private Virtual, Redes Virtuales Privadas.

WAN: Wide Area Network, Red de Área Amplia.

WWW: World Wide Web, Malla Mundial.

REFERENCIAS BIBLIOGRÁFICAS

[1] S. Jha, Mahbub Hassan "Engineering Internet QoS ", A. H. Inc., Ed., 2002.

[2] H. Schulzrinne, S. Casner, R. Frederick "RTP: A Transport Protocol for Real-Time Applications. RFC3550," 2003.

[3] J. Carrallo, "Internet se dirige a la Tecnología MPLS," in *Revista corporativa de TDE*. vol. 7, 2002.

[4] B. Davie, Y. Rekhter "MPLS technology and applications. ," M. Kaufmann., Ed. San Francisco, 2000.

[5] E. Ibarra García, "MPLS (Multi-Protocol Label Switching) y Calidad de Servicio en Redes IP. ," 2006.

[6] F. A. Paliza, "INGENIERIA EN INTERNET. CALIDAD EN EL SERVICIO," Santa Clara: Universidad Central de Las Villas, 2006.

[7] R. Redford, "Enabling Business IP Services with Multiprotocol Label Switching," 2002.

[8] R. Braden, D. Clark, S. Shenker "Integrated Services in the Internet Architecture: an Overview. RFC1633.," 1994.

[9] J. Babiarz, K. Chan , F. Baker "Configuration Guidelines for DiffServ Service Classes. RFC4594.," 2006.

[10] J. J. Padilla, "IntServ: An approach to Support QoS over IPv6 Networks," in *The Tenth IEEE Symposium On Computers And Communications ISCC.* Cartagena - Spain, 2005.

[11] F. A. Paliza, "Calidad de Servicio. Arquitectura de Servicios Integrados," Santa Clara: Universidad Central de Las Villas, 2007.

[12] L. Zhang, S. Berson, S. Herzog. , "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC2205. ," 1997.

[13] B. Davie, "An Expedited Forwarding PHB (Per-Hop Behavior). RFC3246.," 2002.

[14] J. Heinanen, F. Baker, J. Wroclawski "Assured Forwarding PHB Group. RFC2597.," 1999.

[15] F. A. Paliza, "Calidad de Servicio, Arquitectura de Servicios Diferenciados," Santa Clara: Universidad Central de Las Villas, 2007.

[16] J. M. Angulo, J.R. Hernández, D.A. Moreno, "MPLS (Multi Protocol Label Switching)," 2005.

[17] C. System, "Introduction to MPLS (Session RST-1601)," 2004.

[18] J. Barberá, Telia Iberia, "MPLS: Una arquitectura de backbone para la Internet del siglo XXI.," 2005.

[19] J. Marzo, E. Calle , T. Anjali "QoS Online Routing and MPLS Multilevel Protection: A Survey," in *IEEE Communications Magazine*

2003.

[20] S. Avallone, "An experimental analysis of Diffserv-MPLS interoperability," in *International Conference on Telecommunications*. vol. 1, 2003, pp. 281-287.

[21] L. Faucheur, "Multi-Protocol Label Switching (MPLS): Support of Differentiated Services (IETF RFC 3270)." 2002.

[22] S. Blake, "An Architecture of Differentiated Services. RFC 2475.," 1998.

[23] S. Ganti, "IETF Draft:MPLS Support of Differentiated Services Using E-LSP," 2001.

[24] A. J. Q. Rahul Sawant, "MPLS DiffServ: A Combined Approach," Illinois State University, 2006.

[25] C. System, "MPLS Quality of Service (QoS)," Cisco, Ed., 2007.

[26] C. System, "Quality of Service for Multi-Protocol Label Switching Networks," 2001.

[27] D. O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering over MPLS. RFC 2702.," 1999.

[28] Y.-T. Kim, "Traffic Engineering and Network Management System for QoS-Guaranteed DiffServ Provisioning," in *KNOM Review*. vol. 7, agosto 2004.

[29] C. System, "Cisco IOS Software Configuration Guide (Release 12.2SR)," 2007.

[30] C. System, "Cisco IOS Quality of Service Solutions Command Reference," 2007.

[31] C. System, "Cisco IOS XR Modular Quality of Service Configuration Guide (Cisco IOS XR Software Release 3.7)," 2007.

[32] C. W. Paper, "Cisco IOS MPLS Quality of Service," Cisco, Ed., 2001.

[33] C. W. Paper, "Positioning MPLS," 2002.

[34] C. System, "Configuring Multiprotocol Label Switching on the Optical Services Modules," in *Optical Services Modules Configuration Note*, 2006.

[35] C. System, "Comprehensive Example of a Configured Network Using Cisco IOS-XR Software," 2004.

[36] C. System, "Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.4," 2007.

ANEXOS

Anexo I Tablas de comandos

Tabla 1 Verificación de la política usando el comando show policy-map

Sintaxis	**Descripción**
show policy-map nombre	Visualiza configuración de política
show policy-map interface [nombre[**dlci**| **vc**| **vp**]][{ **input**| **output**} **class** nombre]	Visualiza contadores para una política adjunta a un interfaz, Frame Relay DLCI, ATM PVC, o ATM PVP
show policy-map control-plane [**all**| **slot** valor][{ **input**| **output**}**class** nombre]	Visualiza contadores para una política de control-plane traffic
show policy-map switch-fabric { **unicast**| **multicast** }	Visualiza contadores para una política controlando tráfico enviado a switch fabric

Tabla 2 Criterios Correspondientes usando cabeceras IP y MPLS

Sintaxis	**Criterio Correspondiente**
match access-group { valor| **name** valor }	Lista de acceso numerada o nombrada (Cisco IOS solamente)

Sintaxis	Criterio Correspondiente
match access-group [**ipv4**| **ipv6**]valor	Lista de acceso (Cisco IOS XR solamente)
match precedence lista	Lista de valores de precedencia (IPv4 e IPv6)
match dscp lista	Lista de valores DSCP
match mpls experimental topmost lista	Lista de valores EXP para MPLS
match packet length { **min** valor[**max** valor]|[**min** valor] **max** valor }	Tamaño de paquete IP (incluyendo cabecera)

Tabla 3 Criterios Correspondientes para características Externas de Paquete

Sintaxis	Criterio Correspondiente
match input-interface valor	Paquete de interfaz llegado
match qos-group lista	Lista de marca de clase de paquete interna
match discard-class lista	Lista de marca de paquete interna que identifica perfil de gota

Tabla 4 Criterios Correspondientes para Ethernet, ATM, y Frame Relay

Sintaxis	Criterio Correspondiente

Sintaxis	**Criterio Correspondiente**
match cos lista	Lista de valores de prioridad de usuario Ethernet 802.1Q
match cos inner lista	Lista de valores interiores de prioridad de usuario Ethernet 802.1Q para paquetes con doble encapsulación VLAN
match source-address mac valor	Fuente Ethernet dirección MAC
match destination-address mac valor	Destino Ethernet dirección MAC
match spantree bpdu	Atravesar-árbol de Ethernet BPDU
match vlan rango	Rango de Ethernet VLAN IDs
match vlan inner rango	Rango de VLAN IDs interiores para paquetes con doble encapsulación VLAN
match atm ilmi	Paquetes ATM ILMI
match atm oam	Celdas ATM OAM
match atm clp	Bit ATM CLP
match frame-relay dlci rango	Frame Relay DLCI
match frame-relay de	Bit Frame Relay DE
match frame-relay lmi	Paquetes Frame Relay LMI

Tabla 5 Criterios Correspondientes para Protocolos y Carga útil de Paquete

Sintaxis	**Criterio Correspondiente**

Sintaxis	**Criterio Correspondiente**
match ip rtp start offset	Paquetes RTP con puertos UDP entre start y start+offset
match protocol arp	Paquetes ARP
match protocol cdp	Paquetes CDP
match protocol clns	Paquetes ISO CLNS
match protocol clns_es	Paquetes ISO CLNS ES
match protocol clns_is	Paquetes ISO CLNS IS
match protocol cmns	Paquetes ISO CMNS
match protocol compressedtcp	TCP Comprimido
match protocol ip	Paquetes IPv4
match protocol ipv6	Paquetes IPv6

Tabla 6 Criterios Correspondientes para definiciones de clases jerárquicas

Sintaxis	**Criterio Correspondiente**
match class-map nombre	Nombre de Mapa de clase

Tabla 7 Criterios de marca para paquetes IP y MPLS

Sintaxis	**Criterio Correspondiente**
set precedence valor	precedencia de IPv4 e IPv6
set precedence tunnel valor	precedencia para ser usada por cabecera IP
set dscp valor	DSCP de IPv4 e IPv6
set dscp tunnel valor	DSCP para ser usado por cabecera de túnel IP
set mpls experimental imposition valor	EXP bits to be used by push operation
set mpls experimental topmost valor	Bits EXP en cabecera MPLS

Tabla 8 Criterios para marcar campos internos de dispositivo

Sintaxis	**Criterio Correspondiente**
set qos-group valor	Campo interno para clase del paquete
set discard-class valor	Campo interno para perfil de gota del paquete

Tabla 9 Criterios de marca para Ethernet, ATM, y Frame Relay

Sintaxis	**Criterio Correspondiente**
set cos valor	Prioridad de usuario Ethernet 802.1Q
set atm-clp	Bit ATM CLP
set fr-de	Bit Frame Relay DE

Tabla 10 Correlación entre criterios de marca

Sintaxis	**Descripción**
set para-campo desde-campo [**table** nombre]	Correlación entre dos campos de paquete (por ejemplo, DSCP, precedencia de IP, MPLS EXP, marcas internas, prioridad de usuario 802.1Q)

Tabla 11 Correlaciones declaradas en un mapa de tabla

Sintaxis	**Descripción**
map from valor **to** valor	Declaración que correlaciona dos valores
default { valor|**copy**|**ignore** }	Acción de correlación por defecto

Tabla 12 Controlador de perfil de tráfico de una razón

Sintaxis	**Definición de perfil**
police rate-valor [bc-valor [be- valor]]	Términos absolutos con sintaxis compacta
police cir valor [**bc** valor [**be** valor]]	Términos absolutos con palabras claves
police rate valor [**burst** valor [**peak-burst** valor]]	Términos absolutos con palabras claves (sintaxis alternativa)
police cir percent valor [**bc** valor **ms** [**be** valor ms]]	Relativo a ancho de banda esencial
police rate percent valor [**burst** valor **ms** [**peak-burst** valor ms]]	Relativo a ancho de banda esencial (sintaxis alternativa)

Tabla 13 Controlador de perfil de tráfico de razón doble

Sintaxis	**Definición de perfil**
police cir valor [**bc** valor]**pir** valor [**be** valor]	Términos absolutos
police rate valor [**burst** valor]**peak-rate** valor [**peakburst** valor]	Términos absolutos (sintaxis alternativa)
police cir percent valor [**bc** valor **ms**]**pir percent** valor[**be** valor **ms**]	Relativo a ancho de banda esencial
police rate percent valor [**burst** valor **ms**]**peak-rate percent** valor [**peak-burst** valor **ms**]	Relativo a ancho de banda esencial (sintaxis alternativa)

Tabla 14 Tipos de acción del controlador

Sintaxis	**Resultado con Controlador de una razón**	**Resultado con Controlador de razón doble**
conform-action	Bastante señal en el primer cubo	Bastante señal en ambos cubos
exceed- action	Bastante señal en el segundo cubo solamente	Bastante señal en el segundo cubo solamente
violate-action	Poca señal en ambos cubos	Poca señal en ambos cubos

Tabla 15 Acciones del policer

Sintaxis	**Descripción**
drop	Gota del paquete
transmit	Transmite paquete sin modoficación

Sintaxis	**Descripción**
set-prec-transmit valor	Precedencia de IPv4 e IPv6
set precedence valor	Precedencia de IPv4 e IPv6 (sintaxis alternativa)
set-prec-tunnel-transmit valor	Precedencia para ser usada por operación de túnel IP
set precedence tunnel valor	Precedencia para ser usada por operación de túnel IP (sintaxis alternativa)
set-dscp-transmit valor	DSCP de IPv4 e IPv6
set dscp valor	DSCP de IPv4 e IPv6 (sintaxis alternativa)
set-dscp-tunnel-transmit valor	DSCP para ser usado por operación de túnel IP
set dscp tunnel valor	DSCP para ser usado por operación de túnel IP (sintaxis alternativa)
set-mpls-exp-imposition-transmit valor	Bits EXP para ser usados por operación push
set mpls experimental imposition valor	Bits EXP para ser usados por operación push (sintaxis alternativa)
set-mpls-exp-topmost-transmit valor	Bits EXP en la cabecera MPLS
set mpls experimental topmost valor	Bits EXP en la cabecera MPLS (sintaxis alternativa)
set-qos-transmit valor	Espacio interno para clase de paquete
set qos-group valor	Espacio interno para clase de paquete (sintaxis alternativa)
set-discard-class-transmit valor	Espacio interno para perfil de gota de paquete

Sintaxis	Descripción
set discard-class valor	Espacio interno para clase de paquete (sintaxis alternativa)
set-cos-transmit valor	Prioridad de usuario de Ethernet 802.1Q
set cos valor	Prioridad de usuario de Ethernet 802.1Q (sintaxis alternativa)
set-clp-transmit	Bit ATM CLP
set atm-clp	Bit ATM CLP (sintaxis alternativa)
set-frde-transmit	Bit Frame Relay DE
set fr-de	Bit Frame Relay DE (sintaxis alternativa)

Tabla 16 Definición de Color para color informado del policer

Sintaxis	Descripción
conform-color nombre	Clase asociada con color de conformado
exceed-color nombre	Clase asociada con color de excedido

Tabla 17 Formación media y máxima de paquete

Sintaxis	Descripción
shape average rate-valor [burst]	shaper promedio con la definición de canal de transmisión en términos absolutos e intervalo de formación fijado

Sintaxis	**Descripción**
shape average rate-valor [bc-valor [be-valor]]	shaper promedio con la definición de canal de transmisión en términos absolutos e intervalo de formación configurable
shape peak rate-valor [bc-valor [be-valor]]	Shaper máximo con la definición de canal de transmisión en términos absolutos e intervalo de formación configurable
shape average percent rate-valor [burst]**ms**	V
shape average percent rate-valor [bc-valor ms [be-valor **ms**]]	Shaper promedio con la definición de canal de transmisión relativa a ancho de banda esencial e intervalo de formación configurable
shape peak percent rate-valor [bc-valor **ms**[be-valor **ms**]]	Shaper máximo con la definición de canal de transmisión relativa a ancho de banda esencial e intervalo de formación configurable

Tabla 18 Formación adaptable para Frame Relay

Sintaxis	**Descripción**
shape adaptive valor	Reduce la razón de formación con el arribo de notificaciones de congestión
shape adaptive percent valor	Reduce la razón de formación (relativa al ancho de banda esencial) con el arribo de notificaciones de congestión

Tabla 19 Asignación de ancho de banda y prioridades de tráfico durante la congestión

Sintaxis	**Descripción**
bandwidth valor	Asignación de ancho de banda mínimo

Sintaxis	Descripción
bandwidth percent valor	Asignación de ancho de banda mínimo con relación al ancho de banda esencial
bandwidth remaining percent valor	Asignación de ancho de banda de exceso
priority [**level** valor][rate-valor[burst-valor]]	Priorización de latencia baja con un policer condicional opcional
priority [**level** valor] **percent** [rate-valor[burst-valor]]	Priorización de latencia baja con un policer condicional opcional en una razón relativa al ancho de banda esencial

Tabla20 Tamaño máximo de cola

Sintaxis	Descripción
queue-limit [valor[**packets**|**bytes**|**cells**|**ms**|**us**]	Tamaño máximo de cola

Tabla 21 Tamaño de cola máximo para marcas de paquete específicas

Sintaxis	Campo ponderado
queue-limit precedence valor limit-valor [**packets** | **bytes** | **ms**]	Precedencia de IPv4, precedencia de IPv6, o MPLS EXP
queue-limit dscp valor limit-valor [**packets** | **bytes** | **ms**]	IPv4 DSCP, IPv6 DSCP, o MPLS EXP
queue-limit discard-class valor limit-valor [**packets** | **bytes** | **ms**]	Clase de descarte
queue-limit cos valor limit-valor [**packets** | **bytes** | **ms**]	Prioridad de usuario de Ethernet 802.1Q

Sintaxis	Campo ponderado
queue-limit clp valor limit-valor [**packets** | **bytes** | **ms**]	Bit ATM CLP

Tabla 22 WRED usando diferentes campos ponderados

Sintaxis	Campo ponderado
random-detect precedence-based	Precedencia de IPv4, precedencia de IPv6, y MPLS EXP
random-detect dscp-based	IPv4 DSCP, IPv6 DSCP, y MPLS EXP
random-detect discard-class-based	Espacio interno para el perfil de gota del paquete
random-detect cos-based	Prioridad de usuario de Ethernet 802.1Q
random-detect clp-based	Bit ATM CLP

Tabla 23 Umbrales WRED para marcas específicas de paquete

Sintaxis	Campo ponderado
random-detect precedence rango min-valor [**packets** | **bytes** | **ms**]max-valor [**packets** | **bytes** | **ms** | **us** prob-den-valor]	Precedencia de IPv4, precedencia de IPv6, o MPLS EXP
random-detect dscp rango min-valor [**packets** | **bytes** | **ms**] max-valor [**packets** | **bytes** | **ms** | **us**] prob-den-valor	IPv4 DSCP, IPv6 DSCP, o MPLS EXP
random-detect exp rango min-valor [**packet** | **bytes** | **ms** max-valor [**packet** | **bytes** | **ms** [prob-den-valor]	MPLS EXP

Sintaxis	**Campo ponderado**
random-detect discard-class rango min-valor [**packets** | **bytes cells** | **ms** max-valor [**packets** | **bytes** | **cells** | **ms** probden-valor	Clase de descarte
random-detect cos rango min-valor [**packets** | **bytes** | **ms** | **us** max-valor [**packets** | **bytes** | **ms** prob-den-valor	Prioridad de usuario de Ethernet 802.1Q
random-detect clp valor min-valor [**cells** | **ms**]max-valor [**cellsms** | **us**] prob-den-valor	Bit ATM CLP

Tabla 24 LFI con MLP y fragmentación FRF.12 de Frame Relay

Sintaxis	**Descripción**
ppp multilink interleave	LFI para MLP
ppp multilink fragment {delay valor **size** valor}	Tamaño de fragmento para LFI con encapsulación MLP
frame-relay fragment valor **end-to-end**	Tamaño de fragmento para LFI con encapsulación Frame Relay

Tabla 25 Compresión de cabecera RTP y TCP

Sintaxis	**Descripción**
compression header ip [**rtp** | **tcp**]	Comprima cabeceras RTP/TCP

Tabla 26 Bandwidth Reference for QoS

Sintaxis	**Descripción**

Sintaxis	**Descripción**
bandwidth qos-reference [**input** | **output**] valor	Tamaño máximo de cola

Tabla 27 Unidades por defecto para los comandos del MQC con parámetros de rate y burst

Comandos	**Unidades de Rate**	**Unidades de Burst**
police	Bits por segundo	Bytes
police cir	Bits por segundo	Bytes
police rate	Bits por segundo	Bytes
police cir percent	No aplicable	Milisegundos
police rate percent	No aplicable	Milisegundos
shape average	Bits por segundo	Bits
shape average	Bits por segundo	Bits
shape peak	Bits por segundo	Bits
shape average percent	No aplicable	Milisegundos
shape peak percent	No aplicable	Milisegundos
shape adaptive	Bits por segundo no configurable	
shape adaptive percent	No aplicable	No configurable
bandwidth	Kilobits por segundos	No configurable
bandwidth percent	No aplicable	No configurable

Comandos	Unidades de Rate	Unidades de Burst
bandwidth remaining percent	No aplicable	No configurable
priority	Kilobits por segundos	Bytes
priority percent	No aplicable	Milisegundos

Tabla 28 Unidades por defecto para los comandos del MQC con parámetros de tamaño de cola

Comando	Unidades de tamaño de cola
queue-limit	Paquetes
random-detect	Paquetes

Tabla 29 Unidades de rate configurables en el MQC

Clave del comando	Unidades
bps	Bits por segundos
kbps	Kilobits por segundos
mbps	Megabits por segundos
gbps	Gigabits por segundos
pps	Paquetes por segundo

Clave del comando	Unidades
cps	Celdas ATM por segundo

Tabla 30 Unidades de memoria configurables en el MQC

Clave del comando	Unidades
bytes	Bytes
kbytes	Kilobytes
mbytes	Megabytes
gbytes	Gigabytes
packets	Paquetes por segundo
cells	Celdas ATM

Tabla 31 Unidades de tiempo configurables en el MQC

Clave del comando	Unidades
ms	Milisegundos
us	Microsegundos

Anexo II Marcas por defecto

Tabla 1 Acciones de marca por defecto para MPLS EXP

Operación MPLS	**Acción de marca por defecto**
Push	Poner MPLS EXP en todas las etiquetas impuestas usando marca en cabecera encapsulada (MPLS EXP, Precedencia de IP o prioridad de usuario Ethernet 802.1Q).
Swap	Mantener valor MPLS EXP
Pop	No modificar marca en cabecera expuesta

Tabla 2 Acciones de marca por defecto para túnel IP

Operación de túnel IP	**Acción de marca por defecto**
Tunnel Encapsulation	Poner encabezamiento de túnel DSCP utilizando encapsulado DSCP para IP sobre GRE o encapsulado EXP para MPLS sobre GRE. Para L2TP, poner DSCP por defecto (cero).
Tunnel Decapsulation	No modificar DSCP en cabecera expuesta

Tabla 3 Marca por defecto para Tráfico IP generado localmente

Protocolo	**Marca IP por defecto**
Bidirectional Forwarding Detection (BFD)	CS6
Border Gateway Protocol (BGP)	CS6
Data-link switching (DLSw)	CS5 (TCP puerto 2065) CS4 (TCP puerto 1981)

Protocolo	**Marca IP por defecto**
	CS3 (TCP puerto 1982) CS2 (TCP puerto 1981)
Distance Vector Multicast Routing Protocol (DVMRP)	CS6
Enhanced Interior Gateway Routing Protocol (EIGRP)	CS6
Gateway Load Balancing Protocol (GLBP)	CS6
Generic routing encapsulation (GRE)	CS6
Hot Standby Router Protocol (HSRP)	CS6
Internet Control Message Protocol (ICMP)	CS6
Internet Group Management Protocol (IGMP)	CS6
Layer 2 Tunneling Protocol (L2TP)	CS6
Label Distribution Protocol (LDP)	CS6
Mobile IP (MIP)	CS6
Multicast Source Discovery Protocol (MSDP)	CS6
Next Hop Resolution Protocol (NHRP)	CS6
Network Time Protocol (NTP)	CS6
Open Shortest Path First (OSPF)	CS6

Protocolo	**Marca IP por defecto**
Protocol-Independent Multicast (PIM)	CS6
Rate Based Satellite Control Protocol (RBSCP)	CS6
Router Port Group Management Protocol (RGMP)	CS6
Routing Information Protocol (RIP)	CS6
Resource Reservation Protocol Traffic Engineering (RSVP-TE)	CS6
Stack Group Bidding Protocol (SGBP)	CS6
Secure Shell (SSH)	CS6
Stateful Network Address Translation (NAT)	CS6
Telnet	CS6
Virtual Router Redundancy Protocol (VRRP)	CS6