

**MECANISMOS DE BALANCEO DE CARGA EN MPLS CON
RSVP-TE Y OSPF**



**WILMAR RAFAEL MUÑOZ BRAVO
MARCO ANTONIO TRUJILLO CASTRILLÓN**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES
GRUPO I + D NUEVAS TECNOLOGÍAS EN
TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
POPAYÁN
2006**

**MECANISMOS DE BALANCEO DE CARGA EN MPLS CON
RSVP-TE Y OSPF**



**WILMAR RAFAEL MUÑOZ BRAVO
MARCO ANTONIO TRUJILLO CASTRILLÓN**

Trabajo de Grado para optar el título de
Ingeniero en Electrónica y Telecomunicaciones

Director

Ing. OSCAR J. CALDERON C.

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES
GRUPO I + D NUEVAS TECNOLOGÍAS EN
TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
POPAYÁN
2006**



CONTENIDO

INTRODUCCIÓN	11
CAPITULO I. MPLS E INGENIERÍA DE TRÁFICO	14
1.1 INGENIERÍA DE TRÁFICO	14
1.1.1 Objetivos de Desempeño de la Ingeniería de Tráfico	14
1.1.2 Capacidades de la Ingeniería de Tráfico	16
1.2 CONCEPTOS BASICOS DE LA ARQUITECTURA MPLS	17
1.2.1 Funcionamiento de la Arquitectura MPLS	17
1.2.2 Clase Equivalente de Reenvío	18
1.2.3 Funciones De Control y Reenvío	19
1.3 INGENIERÍA DE TRÁFICO SOBRE MPLS	20
1.3.1 Enrutamiento Explícito	21
1.3.2 Balanceo de Carga	22
CAPITULO II. BALANCEO DE CARGA EN MPLS	23
2.1 OBJETIVO DE BALANCEO DE CARGA	23
2.2 PRINCIPIOS BÁSICOS DE INGENIERÍA DE TRÁFICO	24
2.3 PARÁMETROS QUE SE DEBEN TENER EN CUENTA PARA LA IMPLEMENTACIÓN DE MECANISMOS DE BALANCEO DE CARGA EN REDES MPLS	25
2.3.1 Control	26
2.3.2 Medidas	27
2.3.3 Caracterización	27
2.3.4 Integrar los Procesos de Enrutamiento con los de Conmutación	28
2.3.5 Todos los Anteriores a un Bajo Costo	28
2.4 BALANCEO DE CARGA EN MPLS	28
2.4.1 Clasificaciones del Balanceo de carga en MPLS	30
2.4.2 Esquemas Propuestos en Balanceo de Carga para Control de congestión en MPLS	30
2.4.2.1 Mecanismos Basados en Enrutamiento o CBR	31
2.4.2.1.1 Topology-based Static Load Balancing Algorithm (TSLB)	31
2.4.2.1.2 Resource-based Static Load Balancing Algorithm (RSLB)	31
2.4.2.1.3 On-demand Optimization of Label Switched Paths in MPLS Networks	32
2.4.2.1.4 MIRA (Minimun Interference Routing Algorithm)	32
2.4.2.2 Problemas comunes en los Mecanismos CBR	32
2.4.2.3 Mecanismos Basados en Reparto o División de Tráfico	34
2.4.2.3.1 MATE (MPLS Adaptive Traffic Engineering)	34
2.4.2.3.2 Ejecución Rápida de Ingeniería de Tráfico (FATE: Fast Acting Traffic Engineering)	36
2.4.2.3.3 Dynamic Load Balancing Algorithm (DLB)	36
2.4.2.3.4 Delay-based Adaptive Load Balancing Algorithm	37
2.4.2.3.5 DYLBAL (DYnamic Load Balancing Algorithm)	37



2.4.2.3.6 LCM (Load balance with Congestion and Mean utilization thresholds)	38
2.4.3 Requerimientos Básicos de los algoritmos de división de tráfico	39
CAPÍTULO III. ANÁLISIS DEL PROTOCOLO RSVP-TE	41
3.1 PROTOCOLO RSVP	41
3.2 RSVP-TE: EXTENSIONES DE RSVP PARA TÚNELES LSP	42
3.2.1. Túneles LSP y Túneles TE	43
3.2.2 Acciones de los Túneles LSP	44
3.2.2.1 Capacidad para establecer túneles LSP con o sin requerimientos de QoS	44
3.2.2.2 Capacidad para reenrutar dinámicamente un túnel LSP establecido	44
3.2.2.3 Capacidad para observar la ruta actual que atraviesa un túnel LSP establecido.	45
3.2.2.4 Capacidad para identificar y diagnosticar túneles LSP	45
3.2.2.5 Capacidad para atribuirse un túnel LSP establecido, bajo control de políticas administrativas	45
3.2.2.6 Capacidad para realizar asignación y distribución de etiquetas downstream-on-demand.	46
3.2.3 Reenrutamiento de Túneles TE	47
3.3 NUEVOS OBJETOS EN RSVP-TE	48
3.4 PARÁMETROS RELEVANTES DE RSVP-TE EN MECANISMOS DE BALANCEO DE CARGA	49
CAPÍTULO IV. ANÁLISIS DEL PROTOCOLO OSPF-TE	52
4.1 OSPFv2	52
4.1.1 Ventajas	52
4.1.2 Desventajas	53
4.1.3 Modo de Operación	53
4.1.3.1 El Algoritmo Link-State	54
4.1.3.2 LSAs (Link State Advertisements)	54
4.1.3.3 Base de Datos de OSPF	55
4.1.3.4 LSAs-Opacos	55
4.2 OSPF-TE	56
4.2.1 LSA-TE	56
4.2.1.1 Carga Útil de un LSA-TE	57
4.2.1.2 Detalles de la carga Útil de un LSA-TE.	57
4.2.1.2.1 TLV "Router Address"	58
4.2.1.2.2 TLV "Link"	58
4.2.2 Elementos de procedimiento	60
4.2.3 Aplicabilidad	60
4.2.4 Limitaciones	61
4.3 OSPF-API	61
4.4 Conclusiones	62
CAPÍTULO V. BALANCEO DE CARGA EN REDES MPLS UTILIZANDO LOS PROTOCOLOS RSVP-TE Y OSPF-TE	63
5.1 CARACTERÍSTICAS DESEABLES QUE EL MECANISMO DE BALANCEO DE CARGA DEBE PRESENTAR	63
5.1.1 Tipo de Control	63



5.1.2 Tipo de Medidas	64
5.1.3 Caracterización	64
5.1.4 Integración de los procesos de enrutamiento con los de conmutación	65
5.1.5 Todos los anteriores a un Bajo Costo	65
5.2 ANÁLISIS FUNCIONAL DE LOS MECANISMOS DE BALANCEO DE CARGA EXISTENTES EN REDES MPLS	65
5.2.1 Análisis Funcional de los mecanismos CBR	65
5.2.2 Análisis Funcional de los Mecanismos Divisores de Tráfico	66
5.2.2.1 Análisis Funcional de MATE	66
5.2.2.2 Análisis Funcional de FATE	67
5.2.2.3 Análisis Funcional de DLB	68
5.2.2.4 Análisis Funcional de Delay-based Adaptive Load Balancing Algorithm	68
5.2.2.5 Análisis Funcional de DYLB	69
5.2.2.6 Análisis Funcional de LCM	69
CAPITULO VI. IMPLEMENTACIÓN Y VALIDACION DEL MECANISMO DLCCM	70
6.1 IMPLEMENTACIÓN DEL MECANISMO DLCCM	70
6.1.1 Topología de Red	70
6.1.2 División Funcional del Mecanismo DLCCM	72
6.1.2.1 Función de Colección	73
6.1.2.2 Función de Distribución	73
6.1.2.3 Función de División y Clasificación	75
6.1.2.4 Función de Asignación	76
6.1.2.4 Función de Balanceo	76
6.1.2.5 Función de Control	77
6.2 ALGORITMO QUE UTILIZA EL MECANISMO DLCCM	78
6.2.1 Funcionamiento del Algoritmo LCM	78
6.2.2 Algoritmo LCM Modificado	81
6.2.3 Algoritmo LCM Generalizado	82
6.3 VALIDACION DEL MECANISMO DLCCM	85
6.3.1 Pre-Configuración de la Red MPLS Experimental	85
6.3.2 Tipo de Tráfico Utilizado	87
6.3.3 Diseño de las Pruebas	88
6.3.3.1 Escenario 1	89
6.3.3.1.1 Prueba 1A	89
6.3.3.1.1.1 Cronograma de la Prueba 1A	89
6.3.3.1.1.2 Resultados de la Prueba 1A	91
6.3.3.1.1.3 Análisis de Resultados, Prueba 1A	92
6.3.3.1.2 Prueba 2A	93
6.3.3.1.2.1 Cronograma de la Prueba 2A	94
6.3.3.1.2.2 Resultados de la Prueba 2A	95
6.3.3.1.2.3 Análisis de Resultados, Prueba 2A	96
6.3.3.1.3 Prueba 3A	97
6.3.3.1.3.1 Cronograma de la Prueba 3A	97
6.3.3.1.3.2 Resultados de la Prueba 3A	99
6.3.3.1.3.3 Análisis de Resultados, Prueba 3A	100



6.3.3.2 Escenario 2	101
6.3.3.2.1 Prueba 1B	101
6.3.3.2.1.1 Cronograma de la Prueba 1B	102
6.3.3.2.1.2 Resultados de la Prueba 1B	103
6.3.3.2.1.3 Análisis de Resultados, Prueba 1B	104
6.3.3.2.2 Prueba 2B	105
6.3.3.2.2.1 Cronograma de la Prueba 2B	105
6.3.3.2.2.2 Resultados de la Prueba 2B	106
6.3.3.2.2.3 Análisis de Resultados, Prueba 2B	107
6.3.3.2.3 Prueba 3B	107
6.3.3.2.3.1 Cronograma de la Prueba 3A	107
6.3.3.2.3.2 Resultados de la Prueba 3B	109
6.3.3.2.3.3 Análisis de Resultados, Prueba 3B	109
6.4 ANÁLISIS DE RESULTADOS GENERALES	109
6.4.1 Resultados Escenario N° 1.	110
6.4.2 Resultados Escenario N° 2.	111
VII. CONCLUSIONES	112
RECOMENDACIONES Y TRABAJOS FUTUROS	115
REFERENCIAS	116



LISTA DE FIGURAS

Figura 1: Camino más corto IGP vs. Camino TE	16
Figura 2: Intercambio de Etiquetas	18
Figura 3: Funciones de reenvío y control	19
Figura 4: Modelo de referencia para balanceo de carga	29
Figura 5: Modelo funcional para balanceo de carga	29
Figura 6: Ejemplo de topología de red con un enlace cuello de botella	33
Figura 7: El diagrama funcional de MATE	35
Figura 8: Encabezado de un LSA	57
Figura 9: Formato de un TLV	57
Figura 10: Red MPLS Experimental	71
Figura 11: Diagrama en Bloques del Mecanismo de Balanceo de carga	72
Figura 12: Diagrama de Flujos del Algoritmo LCM	79
Figura 13: Diagrama de flujo del algoritmo LCM Modificado	82
Figura 14: Algoritmo LCM Generalizado	84
Figura 15: Escenario 1, Una sola Fuente de Tráfico	89
Figura 16: Tráfico Total Prueba 1A	91
Figura 17: Resultados Prueba 1A	91
Figura 18: Resultados Prueba 2A	95
Figura 19: Tráfico Total Prueba 3A	99
Figura 20: Resultados Prueba 3A	99
Figura 21: Escenario 2, Dos Fuentes de Tráfico	101
Figura 22: Tráfico Total Prueba 1B	103
Figura 23: Resultados Prueba 1B	103
Figura 24: Resultados Prueba 2B	106
Figura 25: Tráfico Total Prueba 3B	108
Figura 26: Resultados Prueba 3B	109



LISTA DE TABLAS

Tabla 1: Nuevos Objetos en RSVP-TE.	49
Tabla 2: características de RSVP-TE en mecanismos de Balanceo de carga.....	49
Tabla 3: Tráfico Prueba 1A	90
Tabla 4: Tráfico Prueba 2A	95
Tabla 5: Tráfico Prueba 3A	98
Tabla 6: Tráfico Prueba 1B	102
Tabla 7: Tráfico Prueba 2B	106
Tabla 8: Tráfico Prueba 3B	108
Tabla 9: Resumen de Resultados, sin Cross traffic en la red.	110
Tabla 10: Resumen de Resultados, con Cross traffic en la red.....	111



LISTA DE ANEXOS

- A.** Código del Mecanismo DLCCM
- B.** Configuración de la red Experimental MPLS Bajo un Ambiente Linux



INTRODUCCIÓN

En los últimos años las exigencias y necesidades de contar con redes que tengan la capacidad de utilizar los recursos de que disponen, de manera más eficiente y efectiva, ha abierto un abanico de áreas de investigación, muchas de ellas relacionadas con la ingeniería de tráfico y con MPLS, una de las tecnologías que hoy por hoy se convierte en pilar de las redes de nueva generación.

La exigencia de las nuevas redes demandan, entre otras, buenas características para problemas de confiabilidad, manejados por estos esquemas de recuperación ante fallas y mecanismos de redundancia en recursos de red; igualmente se busca apalejar los problemas de congestión y subutilización de recursos mediante esquemas de balanceo de carga, los cuales buscan distribuir de la manera más efectiva y eficiente, el tráfico que ingresa a la red entre los diferentes enlaces de salida posibles y acorde a las capacidades de estos.

Muchos de los esquemas actuales de balanceo de carga están concentrados en una repartición fija entre los canales de salida, sin tener en cuenta las capacidades disponibles de los recursos a utilizar, y por otro lado realizados en la mayoría de los casos de forma pre-configurada y estática, lo cual si bien es una solución útil el contexto de mejorar la utilización de los recursos de red, dista un tanto de ser la solución óptima.

El presente proyecto es un paso en la consolidación del conocimiento en el área de las Redes Telemáticas Multiservicio que soportan IP. Específicamente abordan el aspecto relacionado con la ingeniería de tráfico visto desde el balanceo de carga; contribuyen en un buen grado a la obtención de una red óptima desde el punto de utilización de recursos de red.

Por otro lado constituye un avance en la investigación académica al aportar elementos de análisis y síntesis sobre el problema de balanceo de carga en redes MPLS, al considerar y evaluar si es viable la utilización de los protocolos OSPF (Plano de Control) y RSVP (plano de envío) en las decisiones de balanceo de carga con carácter dinámico en redes



MPLS. De otro lado, el presente proyecto determina sin lugar a dudas un marco conceptual y analítico sobre el problema de balanceo de carga y las implicaciones que este tiene en el contexto de las redes actuales, ayudando a comprender cómo estos mecanismos impactan a la utilización óptima de recursos, y cómo pueden contribuir a la mejora en otros problemas de congestión de las redes.

Este trabajo de grado contiene en su parte inicial una descripción de la ingeniería de tráfico, sus objetivos de desempeño y capacidades, además presenta una breve descripción de la tecnología MPLS mostrando sus componentes más importantes. Por último se presentan algunas características que permiten realizar ingeniería de tráfico sobre MPLS, mostrando al balanceo de carga como una forma de dar solución a los objetivos de desempeño planteados en la ingeniería de tráfico.

En el capítulo II, se describe ampliamente el balanceo de carga, sus objetivos, los parámetros que se deben tener en cuenta para la implementación de mecanismos de balanceo de carga en redes MPLS, clasificaciones del balanceo de carga y por último algunos esquemas propuestos en balanceo de carga para control de congestión en MPLS. Con lo anterior se pretende observar un poco el panorama del balanceo de carga en redes MPLS y cómo podría o puede ser implementado en redes de Telecomunicaciones.

En el capítulo III, se realiza un análisis de las extensiones del protocolo RSVP-TE, en la búsqueda de características que permitan desarrollar un mecanismo de balanceo de carga utilizando la tecnología MPLS. Se estudiará los nuevos objetos y las funciones de RSVP-TE para el manejo de Túneles LSP.

En el capítulo IV, se analiza las extensiones realizadas a OSPF para soportar Ingeniería de Tráfico en redes MPLS. Este capítulo al igual que el capítulo III, realiza una búsqueda de las características que permitan desarrollar un mecanismo de balanceo de carga, utilizando la tecnología MPLS.

En el capítulo V, tiene como objetivo determinar las características deseables que debe tener el mecanismo de balanceo de carga a implementar, teniendo en cuenta algunas recomendaciones de la IETF, como también determinar cuáles de los mecanismos de



balanceo de carga analizados en el capítulo 2 permitirían obtener dichas características, y así seleccionar el mecanismo de balanceo de carga que más se aproxime a las características deseadas.

El capítulo VI, se implementa y se valida el mecanismo de balanceo seleccionado (llamado DLCCM). Como primera parte se presentan las principales funciones utilizadas en la implementación del mecanismo DLCCM, seguidamente se da una explicación detallada del funcionamiento del algoritmo de balanceo que utiliza el mecanismo DLCCM. Finalmente se realizan las pruebas necesarias, a fin de validar su funcionamiento.



CAPITULO I. MPLS E INGENIERÍA DE TRÁFICO

1.1 INGENIERÍA DE TRÁFICO

“La Ingeniería de Tráfico (TE: Traffic Engineering) se refiere a la optimización del funcionamiento de las redes operacionales. En general abarca el uso de la tecnología y los principios científicos para la medida, modelamiento, caracterización, control del tráfico de Internet y uso de tal conocimiento y técnicas para lograr objetivos específicos de desempeño” [1].

1.1.1 Objetivos de Desempeño de la Ingeniería de Tráfico

Los principales objetivos de desempeño asociados con la ingeniería de tráfico pueden ser clasificados de la siguiente manera:

- Orientados al tráfico.
- Orientados a los recursos.

Los objetivos de desempeño orientados al tráfico incluyen los aspectos que mejoran la Calidad de Servicio (QoS: Quality of Service) de los flujos de tráfico. Estos objetivos buscan entre otros: minimización de la pérdida de paquetes, minimización del retardo, maximización del throughput y desarrollo de acuerdos de nivel de servicio.

Los objetivos de desempeño orientados a los recursos incluyen aspectos pertenecientes a la optimización de la utilización de los recursos. La gestión eficiente de los recursos de red es la manera para lograr los objetivos propuestos. En general, es deseable asegurar que los recursos de red no lleguen a estar sobre utilizados y congestionados mientras que otros recursos alternos permanezcan inutilizados. De esta manera la función central de la ingeniería de tráfico es manejar eficientemente el uso del ancho de banda de los recursos de red.



Minimizar la congestión es algo esencial en los objetivos de desempeño orientados a los recursos y orientados al tráfico. El problema de la congestión se tiene en cuenta cuando se presenta en espacios de tiempo prolongados y no en pequeños instantes de tiempo. La congestión típicamente se manifiesta bajo dos escenarios [1]:

- Cuando los recursos de red son insuficientes para atender la demanda de tráfico.
- Cuando los flujos de tráfico son mapeados ineficientemente sobre el ancho de banda disponible; causando que ciertos recursos de red lleguen a estar sobre utilizados mientras otros permanecen subutilizados.

El primer tipo de problema de congestión puede ser solucionado por:

- Expansión de capacidad
- Aplicación de técnicas de control de congestión clásicas.

Las técnicas de control de congestión clásicas intentan regular la demanda de tráfico a fin de que éste se acomode sobre los recursos disponibles. Técnicas clásicas para control de congestión incluyen: limitación de las tasas, control de flujo de ventana, gestión de las colas en los enrutadores, control programado y otros.

El segundo tipo de problema de congestión, resulta de la asignación ineficiente de recursos, el cual puede ser solucionado mediante técnicas de ingeniería de tráfico. La ineficiencia en la asignación de recursos puede ser mejorada mediante la implementación de políticas de balanceo de carga. El objetivo de tales estrategias son: minimizar el máximo nivel de congestión o, de forma alterna minimizar la máxima utilización de los recursos [2].

Cuando la congestión es minimizada mediante la eficiente asignación de los recursos, la pérdida de paquetes y los retardos de transmisión disminuyen, e incrementa el Throughput; aumentando por defecto la calidad de servicio percibida por los usuarios finales.

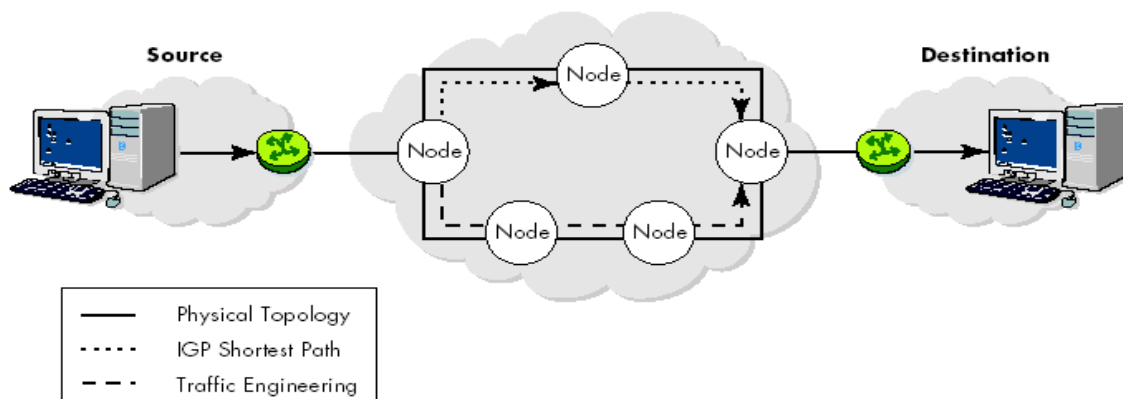
1.1.2 Capacidades de la Ingeniería de Tráfico

La Ingeniería de Tráfico proporciona las capacidades para [3]:

- Enrutar los caminos primarios alrededor de los cuellos de botella o puntos de congestión encontrados en la red.
- Suministrar control preciso sobre cómo el tráfico se reenruta cuando la ruta primaria se enfrenta con fallas simples o múltiples.
- Proporcionar un uso eficiente del ancho de banda disponible, asegurando que los subgrupos de la red no lleguen a estar sobreutilizados mientras otros subgrupos de la red que podrían ser caminos alternos permanezcan subutilizados.
- Mejorar las características de desempeño del tráfico de la red, minimizando la pérdida de paquetes, minimizando los periodos prolongados de congestión y maximizando el throughput.
- Mejorar estadísticamente el desempeño de las características de la red (tales como proporción de pérdida, variación de retardo y transferencia de retardo) requeridas para soportar una Internet multiservicios.

La TE da a los operadores de red un control preciso sobre la ubicación de flujos de tráfico dentro de sus propios dominios de enrutamiento, llamado Sistemas Autónomos (ASs: Autonomous Systems). Esto permite a los operadores llevar los flujos de tráfico fuera del camino más corto seleccionado por el protocolo IGP, sobre un camino físico menos congestionado a través del AS, como se muestra en la figura 1.

Figura 1: Camino más corto IGP vs. Camino TE [4].





Los operadores de red pueden usar la TE para balancear la carga de tráfico sobre varios de sus enlaces, enrutadores y switches, a fin de que ninguno de estos componentes esté sobreutilizado o subutilizado. Este balance de carga les permite explotar la economía de ancho de banda que ha sido suministrada a través de toda la red [4].

A fin de hacer efectiva la Ingeniería de Tráfico, la Internet Engineering Task Force (IETF) presentó la tecnología de Conmutación de Etiquetas Multi-Protocolo (MPLS: Multi Protocol Label Switching) [2] como una tecnología para unificar las diferentes soluciones que los distintos fabricantes estaban proponiendo. Está basado en el uso de etiquetas que identifican la ruta para encaminar los paquetes.

1.2 CONCEPTOS BASICOS DE LA ARQUITECTURA MPLS

MPLS surgió en los últimos años de la década de los 90 como una arquitectura que permitiría mejorar el rendimiento de las redes IP. Sin embargo, actualmente su interés radica en sus capacidades para implementar redes privadas virtuales, funciones de Ingeniería de Tráfico y soporte a QoS sobre Protocolo de Internet (IP: Internet Protocol) [5].

MPLS significa Conmutación de Etiquetas Multiprotocolo. Multiprotocolo porque puede ser utilizado con cualquier protocolo de capa de red y de capa de enlace, y Conmutación de Etiquetas porque los enrutadores intercambian etiquetas (que están anexas a los paquetes) a lo largo de la ruta que los paquetes etiquetados deben recorrer. MPLS es un protocolo orientado a conexión ya que antes de empezar a enviar información traza el o los caminos por los cuales serán enviados los paquetes [6].

1.2.1 Funcionamiento de la Arquitectura MPLS

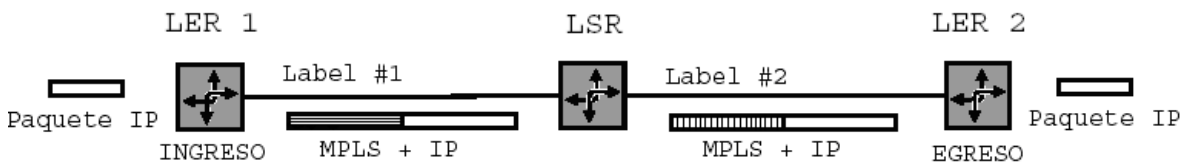
En MPLS la transmisión de datos ocurre a lo largo de circuitos virtuales llamados Caminos de Conmutación de Etiquetas (LSPs: Label Switched Paths). Estos circuitos virtuales constan de una serie de etiquetas impuestas a los paquetes de información que los diversos enrutadores van intercambiando a lo largo del LSP. Las etiquetas utilizadas en los caminos a seguir son elegidas utilizando Protocolos de Distribución de Etiquetas dinámicos como (LDP: Label Distribution Protocol) (RFC 3036), Enrutamiento basado en



Restricciones del Protocolo de Distribución de Etiquetas (CR-LDP: Constraint-based Routed Label Distribution Protocol), Extensiones de RSVP para Túneles LSP (RSVP-TE: RSVP tunnel Extensions) (RFC 3209) o pueden ser configuradas explícitamente.

La arquitectura de una red MPLS está definida en el RFC 3031. Los nodos que intervienen en la formación de un LSP se pueden clasificar como Enrutadores de Etiquetas de Frontera (LERs: Label Edge Routers) y Enrutadores de Conmutación de Etiquetas (LSRs: Label Switching Routers). El LER es el primer enrutador donde los paquetes de información ingresan al LSP y se les coloca una etiqueta, y el último enrutador donde los paquetes emergen del LSP y se les extrae la etiqueta; estos también son llamados LER de ingreso y de egreso respectivamente. Los LSRs son los enrutadores intermedios encargados de intercambiar las etiquetas de los paquetes y reenviar los paquetes a lo largo del LSP. Además, dados dos enrutadores pertenecientes a un mismo LSP se denomina enrutador ascendente (upstream) a aquel que se encuentra más cercano al ingreso y enrutador descendente (downstream) a aquel que está más lejos del ingreso. La figura 2 muestra dichos conceptos.

Figura 2: Intercambio de Etiquetas



1.2.2 Clase Equivalente de Reenvío

La Clase Equivalente de Reenvío (FEC: Forwarding Equivalence Class) es la representación de un grupo de paquetes que comparten los mismos requerimientos de reenvío. El concepto de FEC es local a cada enrutador, cuando un grupo de paquetes en un nodo comparten la subred de destino (prefijo de dirección de destino común) y requerimientos de servicio, pertenecen a la misma FEC. En MPLS la FEC se determina en la entrada y no en cada enrutador. Paquetes con distinto destino pueden agruparse en la misma FEC. Posibles criterios de FEC son: IP Destino, IP Origen - IP Destino, IP Origen - IP Destino - puerto origen - puerto destino, etc. Cada FEC se mapea en un LSP o en un



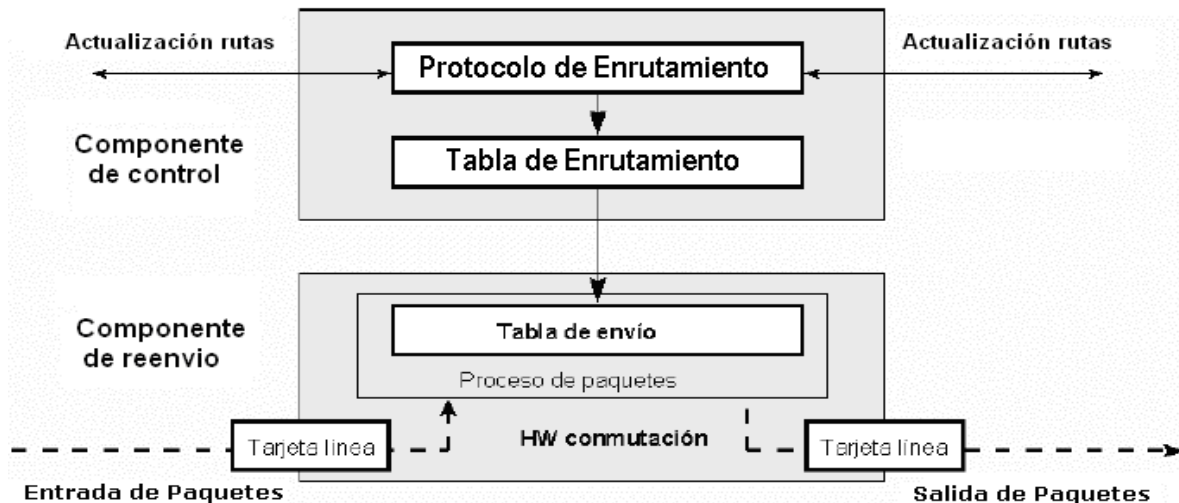
conjunto de ellos si se utiliza enrutamiento multi-camino. Aquí se asigna una etiqueta por cada LSP posible de la FEC.

A cada FEC se le asocia una etiqueta en el nodo de ingreso y se envía el paquete al nodo siguiente con la etiqueta correspondiente. La etiqueta es local, es decir que es un valor arbitrario acordado entre dos vecinos en el camino de un paquete. Cada nodo interior leerá la etiqueta cuando ingresa el paquete y en función de ella (o de ella y la interfaz de entrada) definirá la interfaz de salida y la etiqueta con que se enviará el paquete al próximo nodo. El último nodo de la red, eliminará la etiqueta y enviará el paquete hacia fuera de la red MPLS tal como arribó al dominio. Puede acordarse también entre el último y el penúltimo nodo que sea el penúltimo quien retire la etiqueta, con lo cual puede evitarse en el último nodo dos búsquedas en las tablas de envío (primero en la tabla de MPLS y luego en IP) [5].

1.2.3 Funciones De Control y Reenvío

Se pueden distinguir claramente dos componentes dentro de la arquitectura MPLS: el componente de control y el componente de reenvío (Ver Figura 3).

Figura 3: Funciones de reenvío y control [7]



El componente de control utiliza los protocolos estándar de enrutamiento, como OSPF y el Protocolo de Pasarela de Borde (BGP: Border Gateway Protocol). Utilizando estos



protocolos los enrutadores intercambian información de enrutamiento para construir y mantener las tablas de enrutamiento. Además el componente de control debe crear las asociaciones entre etiquetas y clase de equivalencia de reenvío (FEC: Forwarding Equivalence Class) y distribuir esta información [8].

El componente de reenvío envía los paquetes desde la entrada hacia la salida. Para reenviar los paquetes, el componente de reenvío examina la información de la cabecera del paquete, busca en la tabla de envío la entrada correspondiente y reenvía el paquete. Por tanto, el componente de reenvío consiste en el conjunto de procedimientos que usa el enrutador para tomar la decisión sobre el reenvío de un paquete. Estos algoritmos definen la información del paquete que utiliza el enrutador para encontrar una entrada en la tabla de enrutamiento, así como los procedimientos exactos que el enrutador utiliza para encontrar la entrada [8].

Cada enrutador de la red implementa ambos componentes. Se podría ver el enrutamiento del nivel de red como una composición de ambos componentes (control y reenvío) implementada de una manera distribuida por el conjunto de enrutadores que conforman la red [8].

1.3 INGENIERÍA DE TRÁFICO SOBRE MPLS

MPLS tiene muchos componentes que hacen atractivo su uso en Ingeniería de tráfico. Algunas características para realizar ingeniería de tráfico sobre MPLS son [9]:

- MPLS tiene la habilidad de establecer LSPs que siguen caminos diferentes a los que siguen los protocolos de enrutamiento y algoritmos de reenvío (forwarding).
- Los recursos de red pueden ser reservados dinámicamente con el establecimiento de LSPs y actualizados dinámicamente de acuerdo a las necesidades de los LSPs, a fin de que en los flujos de tráfico se pueda garantizar un nivel de calidad de servicio. El enrutamiento de los LSPs puede ser hecho dependiendo de requerimientos de recurso.
- El tráfico puede ser dividido sobre LSPs en paralelo. Es decir, múltiples LSPs se pueden establecer entre una fuente y un destino final, y el tráfico puede ser distribuido entre los LSPs de acuerdo a un algoritmo. Los LSPs en paralelo pueden tomar muchos caminos diferentes a través de la red.



- Los recursos de la red pueden ser manejados automáticamente con el establecimiento de nuevos LSPs para conocer la demanda inmediata de la red, con recursos libres, como LSPs viejos que ya no son requeridos.
- Se pueden definir procedimientos mejorados, describiendo cómo el tráfico puede ser transferido a LSPs alternos en caso de fallas, e indicando cómo y cuándo los LSPs de respaldo o sustitutos deben entrar en operación.

Dentro de las funciones que permite la ingeniería de tráfico sobre MPLS están:

- Establecer rutas explícitas.
 - Orientar el tráfico hacia donde hay más recursos.
 - Balanceo de carga.
- Obtener estadísticas del uso de un determinado LSP.
 - Análisis de los “cuellos de botella”.
 - Proyectar la futura expansión de la red.
- Enrutamiento Basado en Restricciones (CBR).
 - Permite determinar el LSP a seguir en función de los atributos del tráfico, los recursos de la red y la topología.
 - Garantizar QoS.
- Capacidad de protección ante fallas.
- Generar VPN's.

El desarrollo de este trabajo busca realizar Ingeniería de tráfico por medio de mecanismos de balanceo de carga, por ello se busca estudiar las diferentes maneras de realizar balanceo de carga utilizando la tecnología MPLS, ya que esta tecnología permite en una de sus acciones realizar dicha tarea.

1.3.1 Enrutamiento Explícito

El surgimiento de MPLS con su eficiencia para poder crear caminos explícitos entre dos nodos de una red es un mecanismo fundamental para poder realizar ingeniería de tráfico.



El enrutamiento explícito nos permite que un flujo de datos específicos vaya al destino por un camino predeterminado y no por un camino que se construye paso a paso (hop by hop) como ocurre con OSPF o IS-IS.

La principal característica de MPLS que permite realizar TE es el enrutamiento explícito. Una ruta explícita es una secuencia de nodos lógicos entre un nodo de ingreso y uno de egreso que se definen y establecen desde un nodo de frontera. Una ruta explícita puede ser una lista de direcciones IP. También pueden especificarse los primeros N saltos y luego la ruta definida por el protocolo de enrutamiento IP. Puede usarse también en una ruta explícita el concepto de Nodo Abstracto: Colección de nodos presentados como un solo paso en una ruta explícita. Un ejemplo de nodo abstracto puede ser un Sistema Autónomo.

Si el nodo de ingreso quiere establecer una ruta que no sigue el camino que sigue por defecto el protocolo de enrutamiento IP, debe utilizar un protocolo de distribución de etiquetas que soporte la definición de rutas explícitas. Existen dos definidos por el IETF: CR-LDP y Protocolo de Reserva de Recursos (RSVP: Resource reSerVation Protocol).

1.3.2 Balanceo de Carga

El RFC 3031 [2], habilita la posibilidad de realizar balanceo o reparto de carga entre diferentes LSPs. Balancear carga es un potente mecanismo de ingeniería de tráfico en MPLS. Este mecanismo brinda la posibilidad de enrutar troncales cuyo tráfico es superior a las posibilidades de un único camino en la red, y permite también mejorar el uso de recursos de la red.

Para realizar reparto de carga se deben tener en cuenta dos aspectos. El primero es el algoritmo con el cual se decide los coeficientes de reparto de carga entre los LSPs. El segundo es el mecanismo (una vez fijados los coeficientes) que se utiliza para asignar los paquetes a uno u otro LSP.

Los mecanismos para realizar el reparto de carga, no han sido aún estandarizados por el IETF. Teniendo en cuenta esto, el presente trabajo investiga sobre este tema y pretende hacer un aporte que brinde una solución al problema del reparto de carga en MPLS.



CAPITULO II. BALANCEO DE CARGA EN MPLS

El balanceo de carga es uno de los componentes de la Ingeniería de tráfico, el cual plantea la posibilidad de hacer una óptima utilización de los recursos de la red, evitando así la congestión y aumentando el rendimiento. Las redes IP convencionales, las cuales utilizan la ruta más corta para alcanzar su destino, hacen difícil la implementación de mecanismos que permitan balancear la carga, debido a que no tienen en cuenta rutas alternas que lleguen a un mismo destino [9].

2.1 OBJETIVO DEL BALANCEO DE CARGA

El balanceo de carga es implementado en redes de telecomunicaciones principalmente para afrontar problemas relacionados con la congestión, entendiéndose por congestión el estado de un recurso de red en el cual el tráfico incidente en el recurso excede su capacidad de tráfico de salida, para un intervalo de tiempo determinado [10]. En otros casos se implementan esquemas de balanceo de carga para utilizar de forma equitativa los recursos de la red, de esta manera aunque el objetivo principal no es evitar la congestión, este tipo de esquemas en cierta medida atacan el problema de congestión siempre que existan recursos alternos que permitan balancear carga de tráfico y las capacidades de los enlaces a balancear sean iguales.

Uno de los problemas más críticos que afrontan los Proveedores de Servicio de Internet (ISP: Internet Service Provider), es como minimizar la congestión en sus enlaces, ya que el tráfico en Internet es muy variante y existen las llamadas horas pico para las cuales no es fácil adaptar los recursos de red a la demanda de tráfico. Es por esta razón que para el presente estudio de mecanismos de balanceo de carga en MPLS, se tendrá como principal objetivo abordar el problema de la congestión y de esta manera implícitamente aumentar la QoS en la red.

En redes de conmutación de paquetes la congestión está muy relacionada con los retardos que sufren los paquetes cuando se desplazan de un origen a un destino, reduciendo de esta manera la calidad de servicio y carga máxima posible en la red. En redes Ethernet una carga del 30% aumenta notablemente el número de colisiones las



cuales reducen el ancho de banda disponible en la red. En redes basadas en conmutación de circuitos, la reducción de la congestión implica tener ancho de banda disponible para futuros requerimientos de conexiones o para aumentar la capacidad de reacción de los esquemas implementados para el restablecimiento de los enlaces una vez la falla se ha presentado en la red [11].

2.2 PRINCIPIOS BÁSICOS DE INGENIERÍA DE TRÁFICO

Una sutil, pero práctica ventaja de una sistemática aplicación de los conceptos de ingeniería de tráfico a redes operacionales, es que ayudan a identificar y estructurar los objetivos y las prioridades en términos de aumentar la calidad de servicio entregada a los usuarios finales. La aplicación de los conceptos de ingeniería de tráfico también ayuda en la medida y análisis de los objetivos que se plantean cuando se implementa ingeniería de tráfico [10].

Los objetivos de optimización de ingeniería de tráfico se pueden lograr mediante la administración de la capacidad y la administración del tráfico. La administración de la capacidad incluye planeación, control de enrutamiento, y administración de recursos. Es de particular interés para la ingeniería de tráfico los recursos de la red, los cuales poseen parámetros relevantes a considerar: el ancho de banda de los enlaces, el espaciamiento en el buffer y la capacidad de recursos computacionales de cada uno de ellos. La administración de tráfico incluye funciones de control de tráfico nodal, tales como restricciones de tráfico, administración del encolamiento, planificación y otras funciones que regulan el flujo de tráfico a través de la red o que intervienen en el acceso a recursos de la red entre diferentes paquetes o entre diferentes flujos de tráfico [10].

Un aspecto crítico en la ingeniería de tráfico es la evaluación del desempeño de la red, para lo cual se deben ejecutar procesos de monitoreo y verificación conforme a los objetivos planteados, determinando la efectividad de los mecanismos de Ingeniería de Tráfico implementados. Los resultados del desempeño pueden ser utilizados para identificar problemas existentes, determinar una re-optimización en la red y ayudan a predecir problemas que podrían presentarse [10].

Una evaluación aproximada del desempeño de la red podría ser realizada de diferentes



maneras. Las técnicas más utilizadas incluyen metodologías analíticas, simulaciones y métodos empíricos de medición. Cuando son utilizadas metodologías analíticas o simulaciones, los nodos y los enlaces pueden ser modelados para capturar características relevantes de operación tales como: topologías, ancho de banda, espaciamiento del buffer y políticas de servicio en los nodos (ej. planificación de los enlaces, priorización de paquetes, administración del buffer, etc.). Los modelos analíticos pueden ser usados para bosquejar dinámicamente las características del tráfico, tales como alteraciones considerables en la tasa de transferencia, distribución estadística y adiciones de tráfico en nodos intermedios.

La simulación puede ser utilizada para evaluar el desempeño de las redes o para validar aproximaciones analíticas, sin embargo este mecanismo puede resultar computacionalmente costoso y puede no siempre proveer información suficientemente precisa. Una apropiada aproximación a la evaluación del desempeño de la red se debe hacer involucrando metodologías híbridas, en las cuales se haga uso de técnicas analíticas, simulaciones y métodos empíricos [10]. Las metodologías empíricas han cobrado gran relevancia con el auge que han presentado los proyectos de software libre, con ello se ha visto por parte de los constructores de hardware para telecomunicaciones una gran oportunidad para impulsar proyectos de investigación sobre plataformas libres, que luego de ser depurados son llevados a producción en entornos empresariales, siendo esta metodología una de las más utilizadas en los últimos tiempos [12] [13] [14].

2.3 PARÁMETROS QUE SE DEBEN TENER EN CUENTA PARA LA IMPLEMENTACIÓN DE MECANISMOS DE BALANCEO DE CARGA EN REDES MPLS

Los procesos de balanceo de carga deben cumplir con unos requerimientos básicos especificados en el RFC 2702 [15].

- Control.
- Medidas.
- Caracterización de la red.
- Integrar los procesos de enrutamiento con los de conmutación.
- Todos los anteriores a un bajo costo.



2.3.1 Control

Los esquemas que implementan ingeniería de tráfico y puntualmente balanceo de carga, deben tener un auto-control acorde a los cambios que se presenten en la red, esto con el fin de evitar al máximo la intervención humana en dichos procesos.

El RFC 3272 [10], clasifica los esquemas para control de congestión acorde a los siguientes criterios:

Pro-activa Vs Reactiva: En caso de un control pro-activo, se toman acciones preventivas que permitan evitar posibles estados futuros indeseables en las redes. En el caso de un sistema de control reactivo, responde correctivamente a eventos que ya están ocurriendo en la red.

Respuesta en Escala de Tiempo: Pueden ser caracterizadas como de gran escala aquellos procesos que incluyen actualización de la infraestructura de red, los cuales están entre semanas-años. Las funciones de control de enrutamiento están catalogadas en termino medio (milisegundos a días), son aceptables las respuestas de los sistemas de monitoreo de tráfico, distribución y utilización de los recursos que posteriormente proveen retroalimentación a mecanismos de ingeniería de tráfico ya sean on-line o off-line (ej. para el establecimiento o ajuste de algunos LSPs en redes MPLS, para enrutar tráfico de recursos de red que estén presentado problemas de congestión). Finalmente las funciones de procesamiento de paquetes están catalogadas como de escala corta (ej. administración de la tasa de velocidad, administración de colas y tareas programables), las cuales operan en escalas finas de tiempo, cuyo rango está entre pico segundos a milisegundos, estas corresponden a aplicaciones de tiempo real en el comportamiento estadístico del tráfico.

Control de la Oferta Vs Control de la Demanda: El control de la oferta hace referencia a la implementación de políticas para incrementar la capacidad efectiva de ancho de banda disponible, en orden a decrementar la congestión (ej. balanceo de tráfico sobre toda la red), en tanto que el control de la demanda de tráfico implementa políticas que administran el tráfico para evitar problemas de congestión, un ejemplo de ello, son algunos mecanismos de corta escala de tiempo, como RED (Random Early Detection) y



sus variantes. Sin embargo los sistemas de control de congestión desde el lado de la demanda de tráfico no son muy utilizados en Internet y a menudo se utilizan dentro de sistemas autónomos.

2.3.2 Medidas

Las mediciones permiten evaluar el comportamiento de los esquemas de balanceo de carga implementados, además son parte esencial en los procesos de control, de tal manera que se debe seleccionar un adecuado sistema de mediciones acorde a los parámetros que se pretenden optimizar con la implementación de Ingeniería de Tráfico.

Las medias se pueden obtener básicamente de dos maneras, la primera es haciendo uso de mecanismos fuera de línea, es decir que no hacen parte de los recursos de la red, se debe notar que los esquemas que utilizan este sistema de mediciones tienen restricciones temporales, prácticamente determina que el esquema de control a implementar sea de tipo preventivo y que mediante el uso de algún algoritmo permita predecir comportamientos futuros de la red, la implementación de mecanismos reactivos no es muy utilizada con este tipo de medidas, debido a que la información de las mediciones no esta actualiza al momento de ejecutar los procesos de balanceo de carga y de esta manera las decisiones que se tomen por parte de los algoritmos de reparto de carga pueden tener alta probabilidad de ser erráticas. La segunda es utilizando mecanismos de medición en línea, este tipo de mediciones se caracteriza por hacer uso de los recursos de red para determinar el estado actual de los enlaces, rutas, buffer, etc., con este tipo de medidas es posible plantear algoritmos sencillos que hagan modificaciones pequeñas pero precisas y así hacer un reparto de carga acorde al estado “actual” de los enlaces.

2.3.3 Caracterización

La caracterización de las redes es un proceso previo a la implementación de cualquier esquema de ingeniería de tráfico. De una correcta caracterización de una red operacional depende el éxito de los procesos de ingeniería de tráfico; se deben definir con claridad los objetivos de la implementación, a fin de determinar los parámetros que se deben medir tanto para ejecutar los mecanismos de ingeniería de tráfico, como los parámetros que determinan si los mecanismos implementados están alcanzando los objetivos planteados.



En el proceso de balanceo de carga para evitar la congestión, uno de los parámetros más relevantes es el ancho de banda de los recursos de red, adicionalmente se puede evaluar la prioridad de los servicios a través de la red y así garantizar calidad de servicio por parte del ISP acorde a los Acuerdos de Nivel de Servicios (SLA: Service Level Agreement) establecidos con el cliente.

2.3.4 Integrar los Procesos de Enrutamiento con los de Conmutación

Con la integración de los procesos de enrutamiento y conmutación, se pretende abordar ambos paradigmas con sus ventajas y desventajas, a fin de plantear esquemas que permitan unificar las capacidades y potencialidades del plano de control con la rapidez del plano de conmutación, de tal manera que los procesos realizados en un plano sean reflejados en el otro. Las redes IP convencionales tienen sistemas de control de enrutamiento muy avanzados, pero las decisiones que toman para elección de las rutas no tienen en cuenta el nivel inferior (Nivel de Enlace de Datos) y aunque eligen la ruta más corta de un nodo origen a un nodo destino, ésta no siempre es la mejor ruta. Es por ello que se han planteado modelos de red superpuestos a fin de dar solución a este problema. El modelo que más ha tenido acogida y desarrollo en el sector de las telecomunicaciones está definido en los RFCs 3031 y 3032 y es conocido como el estándar MPLS. El modelo de red que plantea MPLS ha posibilitado el estudio de múltiples esquemas de balanceo de carga para controlar el problema de congestión.

2.3.5 Todos los Anteriores a un Bajo Costo

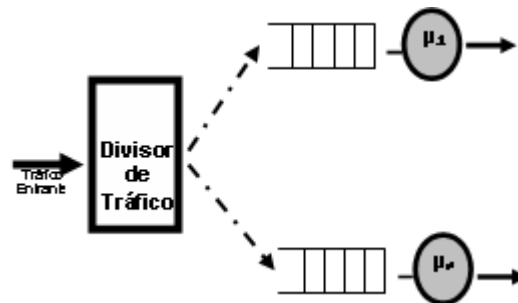
Internet es la suma de múltiples redes interconectadas, hoy día existe un gran despliegue de hardware y software aun no amortizado, por tanto las nuevas tecnologías deben aparte de presentar soluciones óptimas y novedosas a los diversos problemas, ser fácilmente adaptables con la tecnología ya implementada, de tal manera que no deseche los recursos de red actuales y en vez de eso optimicen su desempeño.

2.4 BALANCEO DE CARGA EN MPLS

El Balanceo de Carga (Load Balancing), es un mecanismo importante para mejorar el funcionamiento (en aspectos de caudal, retardo, jitter y pérdidas) y las prestaciones de la

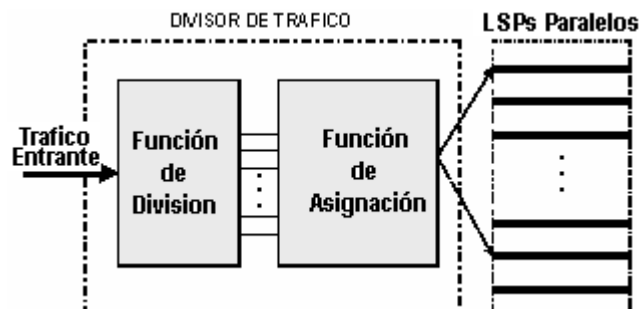
red. Un sistema de balanceo de carga comprende regularmente un Divisor de Tráfico (Traffic Splitter) y múltiples enlaces de salida (Outgoing Links), como se muestra en la Figura 4 [16].

Figura 4: Modelo de referencia para balanceo de carga [17].



De un modo más específico, un mecanismo de balanceo de carga comprenderá funcionalmente dos aspectos (Figura 5).

Figura 5: Modelo funcional para balanceo de carga [17]



La función de división: Es el algoritmo con el cual se decide los coeficientes para repartir el tráfico entrante en porciones adecuadas y acorde a las capacidades de los enlaces de salida y garantizando mantener la secuencia de los paquetes.

La función de asignación: Constituye la parte del mecanismo de balanceo de carga encargada de determinar a cuál LSP y en qué momento entregarle la porción de tráfico que debe transportar [17].



2.4.1 Clasificaciones del Balanceo de carga en MPLS

Una primera clasificación de los mecanismos de balanceo de carga se puede realizar teniendo en cuenta el parámetro de reparto de tráfico que utilizan. En este orden se pueden dividir de forma aproximada en dos grupos principales:

- Basados en Paquetes, se aplican cuando la división de carga trabaja a nivel de paquete, la cual esta bien adaptada a la naturaleza no orientada a conexión de las redes IP.
- Basados en Conexión, donde los flujos de datos son representados con un reducido número de parámetros, y las decisiones de enrutamiento/reenvío afectan a todo el flujo.

El segundo es más complejo de implementar y si no se tienen muchos flujos o estos son muy diferentes, es poco preciso, sin embargo, este último método presenta una fuerte ventaja al mantener el ordenamiento de los paquetes de un flujo [5].

Una segunda clasificación se puede realizar teniendo en cuenta su modo de operación, en este punto se han planteado en entornos MPLS principalmente dos posibles soluciones:

- Mecanismos Basados en Restricciones de Enrutamiento (CBR: Constraint-Based Routing).
- Mecanismos Divisores de tráfico [11].

2.4.2 Esquemas Propuestos de Balanceo de Carga para Control de congestión en MPLS

La mayoría de los esquemas de TE propuestos son de carácter preventivos, ellos realizan la asignación de rutas en la red, en orden a prevenir la congestión [11]. Algunos de los mecanismos que se presentarán continuación definen el algoritmo de balanceo de carga, los parámetros requeridos por el algoritmo de balanceo y el esquema de medidas que se debe utilizar para monitorear dichos parámetros, dichos mecanismos serán referenciados en adelante como mecanismos de balanceo de carga. Aquellos que únicamente definen el



algoritmo de balanceo y los parámetros utilizados por el algoritmo, serán referenciados como algoritmos de balanceo de carga.

2.4.2.1 Mecanismos Basados en Enrutamiento o CBR

Tienen su origen debido a los problemas que presentan los protocolos de enrutamiento en los cuales no se tiene en cuenta la calidad de servicio de los diferentes tipos de tráfico. CBR realiza el cálculo de los LSPs teniendo como argumentos diferentes tipos de restricciones, un ejemplo de ellas pueden ser: el ancho de banda disponible en la red, retardos máximos permisibles y políticas administrativas [11].

2.4.2.1.1 Topology-based Static Load Balancing Algorithm (TSLB)

Este algoritmo es una mejora al algoritmo de camino más corto (Shortest Path). En este algoritmo, un tráfico nuevo se enruta en primer lugar a través del camino más corto; si dicho camino tiene la capacidad suficiente para satisfacer el ancho de banda solicitado por el nuevo tráfico el camino se establece. Si este primer camino no cumple con las necesidades del tráfico entrante se buscará el siguiente trayecto hasta encontrar un camino que satisfaga el requerimiento. Si no se encuentra un camino que cumpla con el requerimiento el algoritmo falla. El principal inconveniente radica en que dado que las fuentes de tráfico transmiten aleatoriamente, un tráfico bajo tomará el camino más corto aunque éste tenga una capacidad muy superior a la requerida, distribuyendo por tanto el tráfico de forma poco razonable y disminuyendo la utilización de los recursos de la red y el throughput [18].

2.4.2.1.2 Resource-based Static Load Balancing Algorithm (RSLB)

En este algoritmo cuando un nuevo flujo de tráfico llega, el nodo de Ingreso seleccionará la ruta cuya capacidad residual este cercana al ancho de banda del flujo de tráfico requerido (pero más grande que el requerido) en el conjunto de rutas disponibles. Así este puede reservar la ruta con capacidad más grande para un futuro flujo de tráfico de entrada que requiera mayor ancho de banda. Puesto que el tráfico cursado en la red es aleatorio, y cada proporción de flujo de tráfico enviada puede ser inconstante o bien fluctuar en un rango específico. Si solo hay un número pequeño de baja proporción de flujos de tráfico



enviándose por un largo tiempo, entonces algunos enlaces de gran capacidad serán inutilizados por ese tiempo. Al mismo tiempo, debido a que el flujo de tráfico está mapeado sobre una ruta en la cual la capacidad residual está muy cercana al ancho de banda requerido, la capacidad del enlace no puede adaptarse bien al flujo de tráfico enviado en la proporción de la fluctuación, y puede resultar en pérdida de datos en un tiempo específico cuando el flujo de tráfico presente picos que excedan la capacidad total del enlace [18].

2.4.2.1.3 On-demand Optimization of Label Switched Paths in MPLS Networks

Jütter et al [19] propone un algoritmo para optimizar el enrutamiento de nuevos LSPs, basado en el re-enrutamiento de un LSP ya establecido y cuando no existan rutas disponibles se debe crear una nueva. La idea de este esquema está basada en los altos niveles de utilización de la red y en CBR para el establecimiento de los LSP.

2.4.2.1.4 MIRA (Minimum Interference Routing Algorithm)

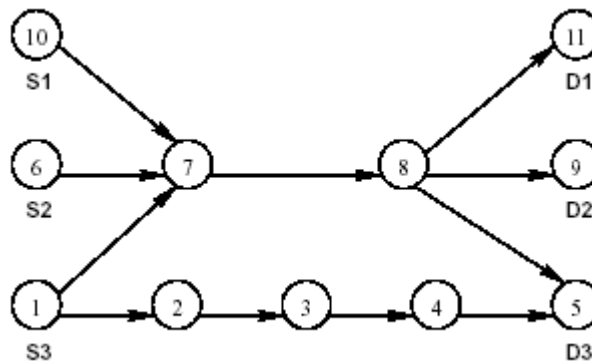
Es uno de los algoritmos CBR más mencionados, está basado en un algoritmo heurístico dinámico para la selección de la ruta. La idea clave pero al tiempo una limitación intrínseca del algoritmo, está en utilizar un conocimiento previo de los pares ingreso-egreso para evitar el enrutamiento sobre enlaces que podrían interferir con futuras selecciones y establecimiento de rutas, MIRA básicamente se ejecuta en tres pasos: primero determina cuáles son los máximos flujos entre todos los pares de ingreso-egreso de un dominio MPLS; como segundo paso determina cuáles enlaces son considerados “críticos” y les asigna pesos prioritarios respecto de otros enlaces, éstos enlaces son identificados por MIRA como enlaces que si eventualmente estuvieran sobrecargados podría ser imposible satisfacer futuras demandas de tráfico entre ellos, hasta tanto no desaparezca la condición de sobrecarga; el tercer paso consiste en correr el algoritmo de Dijkstra haciendo uso de los pesos previamente establecidos para fijar las posibles rutas [20].

2.4.2.2 Problemas comunes en los Mecanismos CBR

Los mecanismos basados en CBR presentan los siguientes problemas:

- Complejidad para identificar los enlaces “críticos”, debido al cálculo del máximo flujo cada vez que un nuevo LSP se establece.
- No pueden estimar cuellos de botella en enlaces que son considerados “críticos” para agrupaciones de nodos. Como lo demostró Wang et al [21], con dos topologías de contra ejemplo, una de ellas se puede observar en la figura 6.
- El algoritmo puede causar desbalance en la utilización de algunos enlaces, puesto que no tiene en cuenta la carga de tráfico para tomar las decisiones de enrutamiento.

Figura 6: Ejemplo de topología de red con un enlace cuello de botella [21]



Considerando el caso de la Figura 6, donde tres pares fuente-destino (S1, D1), (S2, D2), (S3, D3), están conectados por tres rutas ([1-2-3-4-5], [6-7-8-9], [10-7-8-11]) con el mismo ancho de banda residual de una unidad. Cuando llegan tres nuevas demandas de establecimiento de LSPs cada una de una unidad, (S3, D3, 1), (S2, D2, 1), (S1, D1, 1) en el orden respectivo, la primera asignación la realiza el S3 teniendo dos posibles rutas para llegar a su destino D3. La ruta [1-2-3-4-5] tiene un ancho de banda residual de una unidad y la ruta [1-7-8-5] tiene un ancho de banda residual de dos unidades, luego la ruta [1-2-3-4-5], será considerada por MIRA como un enlace “crítico” para el par (S3,D3), por tanto realizará la reserva por el camino [1-7-8-5] dado que este enlace tiene dos unidades residuales de ancho de banda y esta reserva permitiría tener menor “interferencia” para posibles solicitudes de LSPs. La segunda solicitud (S2,D2) es asignada por el camino [6-7-8-9], quedando por tanto este enlace saturado. La tercera solicitud tiene por posible ruta [10-7-8-11] pero dado que el tramo [7-8] ya está saturado, la solicitud será rechazada hasta cuando el tramo saturado tenga ancho de banda residual disponible [21].



2.4.2.3 Mecanismos Basados en Reparto o División de Tráfico

Consecuentemente con lo establecido en el RFC 3272, el reparto de carga se puede catalogar como un mecanismo que intenta hacer control de la congestión desde el lado de la oferta de recursos. Básicamente se encarga de hacer una división del tráfico, balanceando la carga sobre LSPs paralelos aumentando el ancho de banda efectivo en la red. Es utilizado para la asignación adaptativa de tráfico a los enlaces de salida disponibles, dicha asignación se realiza de acuerdo al estado actual de la red; el conocimiento de dicho estado puede estar basado en la utilización del enlace, retardo de paquetes, pérdida de paquetes, etc. Por tal razón, la eficiencia de cualquier mecanismo de balanceo de carga depende crucialmente del proceso de medidas del tráfico en la red y se requiere una gran habilidad para controlarlo de forma precisa, dada la naturaleza dinámica del mismo [22]. Esta herramienta brinda la posibilidad de enrutar troncales cuyo tráfico es superior a las posibilidades de un único camino en la red, y permite también mejorar el uso de recursos de la red.

Generalmente los mecanismos divisores de tráfico son de carácter reactivo y solo unos pocos esquemas para control de congestión se han propuesto. Algunos autores han abordado el problema del reparto de carga entre varios LSPs, siendo una de las herramientas que habilita la arquitectura MPLS para TE. En todos los casos suponen que ya se encuentran configurados un conjunto de LSPs y el problema que se plantea para cada agregado de flujo incidente es: ¿Cómo repartir su carga entre los diferentes LSPs asociados a dicho agregado? En cualquier caso buscan optimizar alguna medida de desempeño de la red, como por ejemplo las pérdidas promedio o el retardo medio en la red.

2.4.2.3.1 MATE (MPLS Adaptive Traffic Engineering)

MATE es un mecanismo de balanceo de carga, cuyo objetivo principal es evitar la congestión en la red mediante el balanceo adaptativo de la carga entre múltiples trayectos, basado en medidas y análisis de la congestión. [6]

Algunas de sus características:

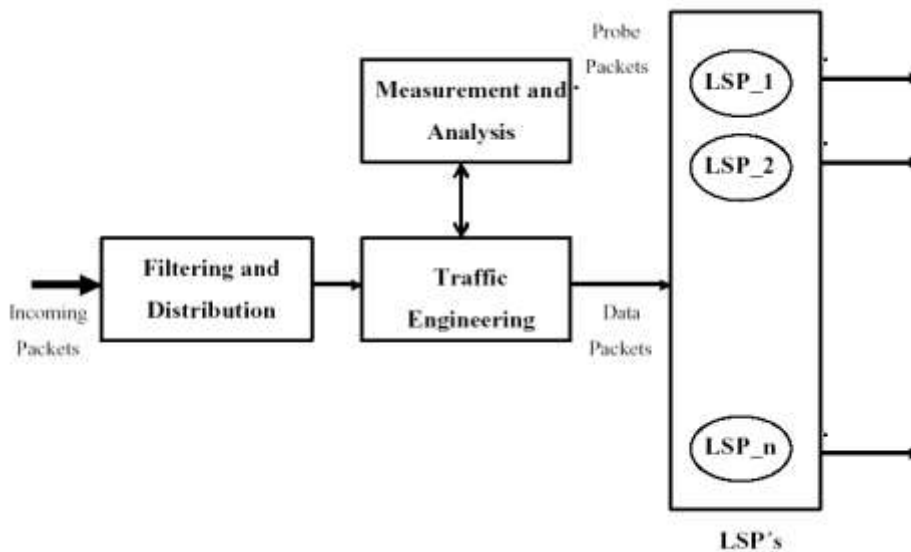
- Control extremo a extremo entre los nodos de ingreso y egreso.



- No se requiere nuevo hardware o protocolo en los nodos intermedios.
- No se requiere conocer la demanda de tráfico.
- Las decisiones de optimización están basadas en la medida de congestión del trayecto.
- Mínimo re-ordenamiento de paquetes.

La red MATE (figura 7) define Engineered Traffic como el tráfico que requiere ser balanceado y Cross Traffic como el tráfico que ingresa a la red a través de los nodos intermedios y sobre los cuales no se tienen ningún tipo de control.

Figura 7: El diagrama funcional de MATE [6].



Estas funciones están ubicadas en el nodo de ingreso de la red MPLS y se describen brevemente a continuación.

- Los paquetes entrantes al nodo de ingreso de la red MPLS llegan al bloque funcional de Filtrado y Distribución, quien es el responsable de distribuir el tráfico entre los diferentes LSP's, de tal forma que se evite que los paquetes que llegan al destino lo hagan fuera de secuencia.
- La función de Ingeniería de Tráfico, consiste de dos fases, Monitorización y Balanceo de Carga. Es responsable de decidir cuándo y cómo se conmuta el tráfico entre los LSP's. Esto se realiza mediante estadísticas, las cuales son obtenidas de las medidas realizadas sobre los paquetes de prueba.



- La función de Análisis y Medidas, es responsable de obtener estadísticas de los LSP's (en un solo sentido) tales como el retardo y la pérdida de paquetes. Esto se logra mediante el envío periódico de paquetes de prueba desde el nodo emisor hasta el nodo receptor y la devolución posterior del mismo [6].

MATE es un mecanismo potente pero asume que el tráfico es constante en media y además es muy sensible a las variaciones del parámetro que se debe medir en línea para ajustar el reparto. Por ejemplo en la propuesta realizada en el trabajo [6] se debe medir la derivada del retardo con respecto a la capacidad de cada enlace y para tal fin se asumen modelos de los enlaces simples. MATE no garantiza requerimientos de QoS al tráfico, simplemente optimiza un costo global de la red, como el retardo promedio, pero nada garantiza que para cada agregado se puedan garantizar sus restricciones de QoS. Se puede tener un retardo promedio mínimo en la red y para algunos agregados no es posible cumplir con sus requerimientos de QoS [6] [23].

2.4.2.3.2 Ejecución Rápida de Ingeniería de Tráfico (FATE: Fast Acting Traffic Engineering)

Holness [24] propone un mecanismo para controlar la congestión en una red MPLS en los cuales el nodo de ingreso (LER: Label Edge Router) y el core (LSR: Label Switched Router) reaccionan con la información recibida de la red, dichos nodos detectan un estado de congestión principalmente de dos maneras: Estimando en los diferentes flujos la cantidad de paquetes perdidos o cuando un requerimiento para la creación de un nuevo LSP no puede ser suministrado debido a que la carga de tráfico en la red ya ha llegado a su tope máximo. Una vez detectado el estado de congestión se hacen los correctivos necesarios para dar solución al problema, por ejemplo, haciendo uso de enrutamiento dinámico de un LSR congestionado a LSRs con baja carga de tráfico.

2.4.2.3.3 Dynamic Load Balancing Algorithm (DLB)

Es un algoritmo de balanceo de carga, el cual parte del conocimiento que el ancho de banda de la ruta, esta dividido en: Capacidad en Uso (UB: Used Bandwidth) y Capacidad Libre (FC: Free Capacity), de tal forma que el algoritmo buscará la ruta que tenga el menor ancho de banda en uso, pero que su capacidad disponible (UB + FC) sea mayor



que la solicitud de ancho de banda del tráfico entrante. DLB toma en cuenta la topología de red y los requerimientos de ancho de banda simultáneamente. En condiciones de baja carga, los flujos de tráfico pueden ser repartidos sobre la ruta disponible más corta y de mayor capacidad; cuando la carga de la red cambie y se torne el tráfico más pesado, los flujos pequeños pueden ser re-enrutados a otras rutas apropiadas y así reservar los enlaces de alta capacidad para los flujos que lo requieran. El inconveniente presentado por este algoritmo se da por el hecho de que los tiempos de re-enrutamiento de los flujos previamente establecidos que se deben acomodar nuevamente deben ser muy rápidos ya que de lo contrario existirán pérdidas de información [18].

2.4.2.3.4 Delay-based Adaptive Load Balancing Algorithm

Este algoritmo de balanceo de carga maneja dos tipos de tráfico, similar como lo hace la red MATE: tráfico de diseño (Engineered Traffic) y tráfico cruzado (Cross Traffic). Los paquetes del tráfico a ser balanceado que ingresan al enrutador de Conmutación de Etiquetas (LSR: Label Switched Router) de Ingreso, son clasificados en N grupos usando un divisor de tráfico basado en una función de hash (hash directo basado en CRC), y posteriormente reparte dichos grupos en LSPs paralelos, de acuerdo a los pesos que cada LSP tenga; dichos pesos son conocidos a partir de los retardos promedio medidos en un solo sentido, utilizando la función EWMA (Exponential Weighed Moving Average). En el esquema CRC16, el divisor de tráfico toma la quintupla (Dirección Origen, Dirección Destino, Puerto Origen, Puerto Destino e Identificador del Protocolo), les aplica CRC y toma el módulo N para obtener el enlace de salida. La función de hash puede ser expresada como:

$$H(.) = \text{CRC16}(5\text{-tupla}) \bmod N$$

Este algoritmo no realiza ningún tipo de ingeniería de tráfico en el core de la red, por tanto las medidas realizadas son extremo a extremo. El parámetro de media que utiliza son los retardos medios que tienen los paquetes que se desplazan de un extremo u otro [22].

2.4.2.3.5 DYLB (DYnamic Load Balancing Algorithm)

Es un algoritmo que implementa una técnica de búsqueda local, donde el proceso básico



es la modificación de la ruta para un único LSP. La idea fundamental que utiliza es la de reenrutar eficientemente LSP's de los enlaces más congestionados. Para los enlaces establecidos se definen procedimientos para monitorear su capacidad y se definen umbrales de congestión. Los LSP's más congestionados son identificados por la mínima capacidad disponible en la red. Para realizar el cálculo de la ruta explícita y ejecutar el algoritmo de balanceo de carga, cada enrutador en la red necesita conocer la topología actual y las capacidades residuales de cada enlace, para determinar e identificar los enlaces más congestionados. Se asume que cada enrutador en la red MPLS ejecuta el algoritmo Estado del Enlace (Link State) con extensiones para conocimiento de ancho de banda residual (Residual Bandwidth Advertisements). Cuando el establecimiento de un nuevo LSP produce la detección de congestión sobre uno de los enlaces de la red (cuando solo se dispone de una cantidad x de ancho de banda residual) el algoritmo de balanceo y el proceso de re-enrutamiento entran en funcionamiento. Un parámetro crítico en el algoritmo es la definición del umbral usado para detectar los enlaces congestionados [18].

2.4.2.3.6 LCM (Load balance with Congestion and Mean utilization thresholds)

El algoritmo de balance de carga con congestión y umbrales de utilización medio (LCM) fue diseñado para mantener la carga del LSP primario dentro de dos niveles marcados por el Umbral de congestión (C: Congestion threshold) y el umbral de utilización medio (M: Mean utilization threshold). Este algoritmo fue publicado y posteriormente fue modificado por sus autores, con el fin de garantizar que la ruta secundaria sea utilizada solo si la carga total excede el nivel de congestión y reducir el tiempo total de estabilización.

El umbral de congestión es fijado al valor de carga más alto aceptable en el enlace, mientras el umbral de utilización medio fija el nivel de retorno de el algoritmo (el punto para enviar de regreso el tráfico de la ruta primaria previamente distribuida por el LSP secundario) y debe ser muy bien afinado por el administrador de red. Como un segundo objetivo del algoritmo LCM la carga del LSP primario debe ser siempre más alta o igual que la carga del LSP secundario [25].

Los mecanismos para realizar el reparto de carga aun no han sido estandarizados por el IETF. Fue propuesto para discusión en el IETF un draft [26] que aborda parte de estos



problemas, pero finalmente fue retirado.

Actualmente en los enrutadores comerciales se puede realizar reparto de carga pero no se puede ajustar en línea los coeficientes de reparto. Es decir, se configuran los coeficientes de reparto y luego estos quedan fijos no existiendo un mecanismo de ajuste si hay variaciones de tráfico. Lo que sucede por defecto, es que si se establece más de un LSP para una troncal, la carga se reparte por partes iguales entre los LSPs establecidos. En algunos casos se implementa una variante algo más sofisticada que es repartir carga inversamente proporcional al ancho de banda reservado por cada LSP. En otros enrutadores se permite configurar coeficientes de reparto de carga y repartir de acuerdo a estos coeficientes. En general también se permite configurar reparto de carga por paquete o por flujo [5].

2.4.3 Requerimientos Básicos de los algoritmos de división de tráfico

Los requerimientos básicos que los esquemas de división de tráfico deben satisfacer para poder realizar balanceo de carga son:

- **Baja Sobrecarga:** Los algoritmos de división de tráfico deben ser muy simples y preferiblemente no tener estados o que estos sean reducidos, pues el procesamiento de cada paquete generará demasiada sobrecarga.
- **Alta Eficiencia:** Una distribución de tráfico muy desigual, puede resultar en una utilización poco uniforme del enlace y en pérdida de ancho de banda.
- **Conservar el orden de los paquetes en los flujos:** Los paquetes mal ordenados dentro de un flujo TCP pueden producir señales de congestión erróneas y hacer que se produzca una degradación del throughput [5].

2.5 Problemas de Balanceo de Carga en redes MPLS

Teniendo en cuenta el estudio realizado a los mecanismos de balanceo de carga, que se plantearon anteriormente, se pueden deducir los principales inconvenientes que se presentan al momento de realizar una posible implementación:



- Cómo obtener el estado actual de la red, es decir Ancho de banda Utilizado, Ancho banda Libre, Ancho de banda Asociado por enlace, etc.
- La alta complejidad que pueden presentar los mecanismos que ejecutan un control proactivo, que se reflejaría en el alto consumo de procesamiento.
- La mayoría de los mecanismos planteados parten del hecho de conocer el ancho de banda de cada nueva conexión, esto es una limitante puesto que esa información solo esta disponible en redes que utilizan protocolos de señalización, que gestionan el ancho de banda específico para cada nueva conexión, con lo cual se agregarían las deficiencias asociadas a las redes que operan bajo reserva de recursos.
- Cómo clasificar el tráfico entrante que se desea balancear.
- Cómo asignar dinámicamente el tráfico, sobre diferentes rutas establecidas.
- Cómo determinar la cantidad de tráfico que debe ser movida de un enlace a otro.



CAPÍTULO III. ANÁLISIS DEL PROTOCOLO RSVP-TE

El presente capítulo pretende analizar las extensiones del protocolo RSVP-TE, en la búsqueda de características que permitan desarrollar un mecanismo de balanceo de carga utilizando la tecnología MPLS. Se estudiará los nuevos objetos y las funciones de RSVP-TE para el manejo de Túneles LSP.

Este capítulo está conformado de la siguiente manera; Primero se describe brevemente las características de RSVP. Luego se presentan algunas extensiones de RSVP para el establecimiento de túneles LSP, explorando las acciones que permite desarrollar RSVP-TE, como lo es el establecimiento de rutas explícitas, el reenrutamiento de LSPs etc. Por último se muestra qué características técnicas brinda el protocolo RSVP-TE para desarrollar un posible mecanismo de Balanceo de carga.

3.1 PROTOCOLO RSVP

El Protocolo de Reserva de Recursos RSVP se utiliza para reservar recursos para una sesión en un entorno de red IP. RSVP pretende proporcionar calidad de servicio estableciendo una reserva de recursos para un flujo determinado. Un host hace una petición de una calidad de servicio específica sobre una red para un flujo particular de una aplicación [8].

Fue desarrollado con el objetivo de hacer que los hosts comuniquen los requerimientos de servicios a la red y los enrutadores puedan establecer un estado de reserva a lo largo de la ruta. Este protocolo permite que varios generadores transmitan a grupos múltiples de receptores, permite que receptores individuales conmuten canales libremente y perfecciona el uso del ancho de banda eliminando al mismo tiempo el congestionamiento.

RSVP opera en el nivel superior de IPv4 o IPv6 ocupando el lugar de un protocolo del nivel transporte según el modelo OSI. Sin embargo, RSVP no transporta datos de aplicación, solamente envía mensajes de señalización para establecer la reserva de recursos.



Características de RSVP:

- En RSVP, los receptores hacen las reservas de QoS.
- La reserva es realizada por flujo.
- RSVP es un protocolo de señalización.
- RSVP debe mantener en cada nodo los requerimientos de reserva.
- RSVP utiliza un conjunto de mensajes de señalización.

RSVP-TE es una extensión del protocolo original RSVP diseñado para ejecutar distribución de etiquetas sobre MPLS, además soporta la creación de rutas explícitas con o sin reserva de recursos. Una de las características adicionales más importante de este protocolo es que permite el re-enrutamiento de los túneles LSP, con el fin de dar una solución ante caídas de red, congestión y cuellos de botella.

3.2 RSVP-TE: EXTENSIONES DE RSVP PARA TÚNELES LSP

Los Hosts y enrutadores que soportan RSVP [27] y MPLS [2] pueden asociar etiquetas con flujos RSVP. Una vez que se establece un LSP, el tráfico a través de la ruta se define con una etiqueta que se aplica en el nodo de ingreso del LSP. A un grupo de paquetes al que se le asignó un mismo valor de etiqueta, pertenecerán a la misma FEC [2] y definirán un “flujo RSVP”. Cuando el tráfico se mapea así sobre un LSP, el LSP es llamado un “túnel LSP”. Cuando las etiquetas se asocian a flujos de tráfico, es posible que los enrutadores identifiquen el estado de reservación apropiado para un paquete, basado en el valor de etiqueta del paquete.

El protocolo de señalización RSVP usa distribución de etiquetas downstream-on-demand. Un requerimiento de etiqueta a un túnel LSP específico, se inicia por un nodo de ingreso a través del mensaje *RSVP Path*. Para este propósito, al mensaje *RSVP Path* se le agrega un objeto *LABEL_REQUEST*. Las etiquetas se asignan descendientemente y se propagan ascendientemente por medio del mensaje *RSVP Resv*, para este propósito el mensaje *RSVP Resv* se extiende con un objeto *LABEL_OBJECT* especial.

El protocolo de señalización RSVP-TE soporta capacidad de enrutamiento explícito. Esto se logra con la incorporación de un objeto *EXPLICIT_ROUTE* a un mensaje *RSVP Path*. El objeto *EXPLICIT_ROUTE* encapsula una concatenación de saltos, los cuales



constituyen el camino de la ruta explícita. Usando el objeto `EXPLICIT_ROUTE` los LSPs pueden ser configurados independientemente del enrutamiento IP convencional. El LSP se puede especificar administrativamente, o se puede calcular automáticamente por una entidad basada en QoS, o por políticas de demanda que tengan en cuenta el estado actual de la red.

3.2.1. Túneles LSP y Túneles TE

De acuerdo a [27], “RSVP define una ‘sesión’ como un flujo de datos con un destino particular y un protocolo a nivel de transporte”. Sin embargo, cuando RSVP y MPLS se combinan, un flujo o sesión puede ser definido con gran flexibilidad y generalidad. El nodo de ingreso de un LSP puede usar una variedad de métodos para determinar a que paquetes se le asigna una etiqueta en particular. Una vez que se asigna una etiqueta a un grupo de paquetes, la etiqueta define el “flujo” a través del LSP. Se hace referencia tanto a un LSP como a un “túnel LSP” porque el tráfico a través de este es transparente para nodos intermedios a lo largo del LSP.

Nuevos objetos `RSVP_SESSION`, `SENDER_TEMPLATE`, y `FILTER_SPEC`, llamados `LSP_TUNNEL_IPv4` y `LSP_TUNNEL_IPv6` han sido definidos para soportar las características de los túneles LSP [28]. La semántica de estos objetos desde la perspectiva de un nodo a lo largo del LSP, es que el tráfico perteneciente a un túnel LSP se identifica solamente en base a paquetes que llegan de un Salto Previo (PHOP: previous hop), con el valor de etiqueta individual asignado por este nodo hacia emisores ascendentes en la sesión. De hecho, la IPv4(v6) que aparece en el nombre del objeto solo denota que la dirección de destino es una dirección IPv4(v6). Cuando nos referimos a estos objetos genéricamente, usamos el calificador `LSP_TUNNEL`.

En algunas aplicaciones se utiliza la asociación de túneles LSP en grupos. Esto puede ser útil durante las operaciones de reenrutamiento o para difundir troncales de tráfico sobre múltiples rutas. En aplicaciones de ingeniería de tráfico tales grupos son llamados Túneles TE (TE tunnels: traffic engineered tunnels). Para habilitar la identificación y asociación de los túneles LSP, se transportan dos identificadores; el túnel ID y el LSP ID. Un túnel ID es parte de un objeto `SESSION_OBJECT`. El `SESSION_OBJECT` identifica a un túnel TE. Los objetos `SENDER_TEMPLATE` y `FILTER_SPEC` transportan al LSP ID.



El objeto *SENDER_TEMPLATE* (o *FILTER_SPEC*) conjuntamente con el objeto *SESSION* identifican inequívocamente a un túnel LSP.

3.2.2 Acciones de los Túneles LSP

A continuación se presentan algunas de las características que soporta RSVP extendidas del RFC 3209 relacionado a las acciones de túneles LSP:

- Capacidad para establecer túneles LSP con o sin requerimientos de QoS.
- Capacidad para reenrutar dinámicamente un túnel LSP establecido.
- Capacidad para observar la ruta actual que atraviesa un túnel LSP establecido.
- Capacidad para identificar y diagnosticar túneles LSP.
- Capacidad para atribuirse un túnel LSP establecido, bajo control de políticas administrativas.
- Capacidad para realizar asignación y distribución de etiquetas downstream-on-demand.

3.2.2.1 Capacidad para establecer túneles LSP con o sin requerimientos de QoS.

Para crear un túnel LSP, el primer nodo MPLS (nodo emisor con respecto a la ruta) crea un mensaje *RSVP Path* con una sesión tipo *LSP_TUNNEL_IPv4* o *LSP_TUNNEL_IPv6* e inserta un objeto *LABEL_REQUEST* dentro del mensaje *Path*. El objeto *LABEL_REQUEST* solicita una etiqueta para dicho *Path* y además indica el protocolo de nivel de red sobre el cual será transportada la ruta.

3.2.2.2 Capacidad para reenrutar dinámicamente un túnel LSP establecido.

Si el nodo emisor conoce una ruta que tiene alta probabilidad de encontrar requerimientos de QoS, o que hace uso eficiente de los recursos de la red, o que satisface algunos requerimientos, el nodo puede decidir usar dicha ruta para alguna o todas las rutas de esta sesión. Para hacer esto, el nodo emisor adiciona un objeto *EXPLICIT_ROUTE* al mensaje *RSVP Path*. El objeto *EXPLICIT_ROUTE* especifica la ruta como una secuencia de nodos abstractos.



Si después de que se ha establecido satisfactoriamente una sesión, el nodo emisor descubre una ruta mejor, éste puede reenrutar dinámicamente la sesión haciendo un simple cambio en el objeto *EXPLICIT_ROUTE*. Si se encuentran problemas con un objeto *EXPLICIT_ROUTE*, a causa de un enrutamiento en lazo cerrado o de que alguna ruta intermedia no lo soporta, se notifica al el nodo emisor.

3.2.2.3 Capacidad para observar la ruta actual que atraviesa un túnel LSP establecido.

Adicionando un objeto *RECORD_ROUTE* al mensaje *Path*, el nodo emisor puede recibir información sobre la ruta que atraviesa un túnel LSP. El nodo emisor puede además usar este objeto para solicitar una notificación de la red, referente a cambios en el enrutamiento. El objeto *RECORD_ROUTE* es análogo al vector *Path*, y por lo tanto puede ser usado para detección de lazos cerrados.

3.2.2.4 Capacidad para identificar y diagnosticar túneles LSP

Finalmente, un objeto *SESSION_ATTRIBUTE* puede ser adicionado al mensaje *Path* para ayudar en la identificación y al diagnóstico de la sesión. Información de control adicional, tal como organización y mantenimiento de prioridades (setup and hold priorities), recursos afines y protección local, también se incluyen en este objeto.

3.2.2.5 Capacidad para atribuirse un túnel LSP establecido, bajo control de políticas administrativas

Los enrutadores a lo largo de la ruta podrían usar el *setup y hold priorities* [28] junto con el objeto *SENDER_TSPEC* y algunos objetos *POLICY_DATA* contenidos en mensajes *Path* como entradas a políticas de control. Por ejemplo, en la aplicación de la ingeniería de tráfico, es conveniente usar el mensaje *Path* para verificar qué ancho de banda existe para una prioridad en particular, a lo largo de una ruta entera, antes de realizar alguna reservación de prioridad más baja. Si se permite que un mensaje *Path* avance cuando no hay recursos suficientes, se realizará un proceso innecesario ya que no se podrá satisfacer el requerimiento de ancho de banda.



3.2.2.6 Capacidad para realizar asignación y distribución de etiquetas downstream-on-demand.

Cuando esta presente un Objeto *EXPLICIT_ROUTE* (*ERO*), el mensaje *Path* se envía hacia su destino a través del camino que especifica el *ERO*. Cada nodo a lo largo de la ruta, graba el *ERO* en su Bloque de Estado de Ruta (*path state block*). El nodo además puede modificar el *ERO* antes de enviar el mensaje *Path*. En este caso el *ERO* modificado se debe almacenar en el *path state block* junto con el *ERO* recibido.

El objeto *LABEL_REQUEST* solicita a los enrutadores intermedios y al nodo receptor que se le suministre una etiqueta para la sesión. Si un nodo es incapaz de proveer la etiqueta, enviara un mensaje *PathErr* "clase de objeto desconocida". Si el objeto *LABEL_REQUEST* no se soporta extremo a extremo, al nodo emisor se le notificará esto por el primer nodo que no proporcionó la etiqueta.

El nodo receptor de un LSP responde a un *LABEL_REQUEST* incluyendo un objeto *LABEL* en su mensaje de respuesta *RSVP Resv*. El objeto *LABEL* es insertado en el *filter spec list* siguiendo al *filter spec* al cual el pertenece.

El mensaje *Resv* se envía ascendentemente hacia el emisor, siguiendo la ruta inversa creada por el mensaje *Path*. Hay que advertir que si la ruta fue creada utilizando un *ERO*, entonces el mensaje *Resv* seguirá el camino inverso creado por el *ERO*.

Cada nodo al recibir el mensaje *Resv* y dentro de él al objeto *LABEL*, lo usa para etiquetar el tráfico saliente y asociarlo al túnel LSP. Un nodo no Emisor debe asignar una nueva etiqueta y ubicarla en el correspondiente objeto *LABEL* del mensaje *Resv*, el cual la envía ascendentemente al PHOP. La etiqueta que se envía ascendentemente en el objeto *LABEL* es la etiqueta que este nodo usará para identificar el tráfico entrante asociado con este túnel LSP.

Un LSP se establece con éxito Cuando el mensaje *Resv* se propaga ascendentemente hacia el nodo emisor.



3.2.3 Reenrutamiento de Túneles TE

Uno de los requerimientos para la Ingeniería de Tráfico es el de reenrutar túneles TE, establecidos bajo ciertas condiciones basadas en políticas administrativas. Por ejemplo, en algunos contextos, una política administrativa podría establecer que un túnel TE se reenrute cuando este disponible una ruta “óptima”. Otro contexto de reenrutamiento de túneles TE, es cuando se presentan fallas en los recursos que atraviesa un túnel TE. Bajo algunas políticas, el reenrutamiento también podría ser necesario cuando se desea retornar un túnel TE a su ruta original, cuando el recurso que ha fallado se reactiva nuevamente.

Cuando el reenrutamiento de un Túnel TE esta en progreso no se busca desestabilizar el tráfico o impactar negativamente las operaciones en las redes. Por tal motivo se debe realizar el reerutamiento de forma adaptable, ya que cuando se crea un nuevo túnel LSP se debe transferir tráfico desde un túnel LSP ya establecido, antes de destruirlo. Este concepto es llamado “conexión antes de desconexión” (*make-before-break*). Un problema se puede presentar cuando un viejo y un nuevo túnel LSP compitan por recursos en un segmento de red que tengan en común. Dependiendo de la disponibilidad de los recursos, esta competencia puede causar Control de Admisión para prevenir que el nuevo túnel LSP se establezca. Una ventaja de usar RSVP para establecer túneles LSP es que soluciona este problema.

Para soportar “conexión antes de desconexión”, entre un viejo y un nuevo LSP, los recursos usados por el viejo túnel LSP no deben ser liberados antes de que el tráfico se transfiera al nuevo túnel LSP, y las reservaciones no se deben contar dos veces ya que esto podría causar que se rechace el nuevo túnel LSP.

Una situación similar puede surgir cuando se desea incrementar ancho de banda en un túnel TE. Una reservación se hace para una cantidad completa necesitada, y no para la diferencia entre el nuevo y el viejo ancho de banda. Si se aplican algunas políticas en nodos intermedios sobre los mensajes *PATH*, un mensaje *PATH* que requiere demasiado ancho de banda podría ser rechazado. En este caso, dependiendo de las políticas locales, si se incrementa el ancho de banda requerido y no se realiza ningún cambio en el objeto *SENDER_TEMPLATE* se puede caer el túnel.



El objeto *LSP_TUNNEL SESSION* combinado con el estilo de reservación *SE (Shared Explicit)* [28] permiten realizar transición de ancho de banda y de enrutamiento. La idea es que el viejo y el nuevo túnel LSP compartan recursos a lo largo del enlace que ellos tienen en común. El objeto *LSP_TUNNEL SESSION* es usado para limitar el alcance de la sesión RSVP en un túnel TE particular. Para identificar un túnel SE TE se necesita conocer la dirección IP de destino (una dirección del nodo que está en el egreso del túnel), un túnel ID, y la dirección IP del nodo del túnel de ingreso, la cual se coloca en el campo *Extended tunnel ID*.

Durante el reenrutamiento u operación de incrementar ancho de banda, un túnel de ingreso necesita funcionar como dos emisores diferentes en la sesión RSVP. Esto se logra incluyendo un "*LSP ID*", el cual se transporta en los objetos *SENDER_TEMPLATE* y *FILTER_SPEC*.

Para realizar reenrutamiento, el nodo de ingreso escoge un nuevo *LSP ID* y forma un nuevo *SENDER_TEMPLATE*. Luego crea un nuevo *ERO* para definir una nueva ruta. Después de eso el nodo envía un nuevo mensaje Path usando el objeto *SESSION* original, el nuevo *SENDER_TEMPLATE* y el *ERO*. Se continúa usando el viejo LSP y se renueva el viejo mensaje Path. Sobre enlaces en común, el objeto *SESSION* compartida y el estilo *SE* permiten al LSP establecerse como un recurso que se comparte con el viejo LSP. Una vez el nodo de ingreso recibe un mensaje *Resv* a causa del nuevo LSP, se puede trasladar tráfico a éste y destruir el viejo LSP [28].

3.3 NUEVOS OBJETOS EN RSVP-TE

En la tabla 1 se muestra el nombre los nuevos objetos que pueden ser adicionados a los mensajes Path y Resv para realizar funciones de ingeniería de tráfico:



Tabla 1: Nuevos Objetos en RSVP-TE.

Nombre del Objeto	Mensaje RSVP Aplicable
<i>LABEL_REQUEST</i>	<i>Path</i>
<i>LABEL</i>	<i>Resv</i>
<i>EXPLICIT_ROUTE</i>	<i>Path</i>
<i>RECORD_ROUTE</i>	<i>Path, Resv</i>
<i>SESSION_ATTRIBUTE</i>	<i>Path</i>

El formato de los anteriores objetos, que soportan RSVP-TE junto con algunos procedimientos de su configuración puede ser observado en el RFC 3209.

3.4 PARÁMETROS RELEVANTES DE RSVP-TE EN MECANISMOS DE BALANCEO DE CARGA

En la tabla 2 se muestran algunas características de las extensiones del protocolo RSVP-TE soportado por el RFC 3209. Estas características son analizadas teniendo en cuenta las variables que un posible mecanismo de Balanceo de carga necesitaría y además muestra si las extensiones de RSVP posibilitan acceder o extraer la información que facilite la implementación de mecanismos de balanceo de carga, a partir de los mensajes y objetos de RSVP-TE.

Tabla 2: características de RSVP-TE en mecanismos de Balanceo de carga.

Características	Soporte
BW-Total	Si
BW-Libre	No
BW- Usado	No
Incremento de BW	Si
Reenrutamiento	Si
Enrutamiento Explícito	Si
Observar Ruta Atravesada	Si
Atribución de Recursos	Si



BW- Total: Esta información puede ser observada en el mensaje *Path*, en el objeto *SENDER_TSPEC*.

BW- Libre: No es posible obtener el ancho de banda libre directamente desde un objeto. Cuando se hace una petición y no existe el ancho de banda requerido, el receptor envía un mensaje de error diciendo que no soporta el ancho de banda requerido.

BW-Usado: No es posible accederlo desde algún objeto.

Incremento de BW y Reenrutamiento: A continuación se describe como se configura un túnel para que sea capaz de mantener la reservación de los recursos (sin que sean contados dos veces) mientras se esta reenrutando o mientras se esta intentando incrementar el ancho de banda. En el mensaje *Path* inicial, el nodo de ingreso forma un Objeto *Sesion*, asigna un *Tunel_ID*, y ubica su dirección IPv4 en el *Extended_Tunnel_ID*. También se forma un *SENDER_TEMPLATE* y se asigna un *LSP_ID*. Después se configura el Túnel de acuerdo al procedimiento normal descrito anteriormente.

Cuando el nodo de egreso recibe el mensaje *Path*, éste responde enviando un mensaje *Resv* con el estilo *SE* hacia el nodo de ingreso.

Cuando un nodo de ingreso que tiene una ruta establecida desea cambiar su ruta, forma un nuevo mensaje *Path*. Usa el Objeto Sesión existente, y no cambia ni el *Túnel_ID* ni el *Extend_Tunnel_ID*. El nodo de ingreso elige un nuevo *LSP_ID* y formar un nuevo *SENDER_TEMPLATE*. Crea un objeto *EXPLICIT_ROUTE* para establecer una nueva ruta. El nuevo mensaje *Path* se envía. El nodo de ingreso actualiza tanto al viejo como el nuevo mensaje *Path*.

Para solicitar un incremento de BW, se crea un mensaje *Path* con un nuevo *LSP_ID*, y se sigue usando el actual *LSP_ID* para asegurar que la reservación no se pierda si la reservación grande falla.

Enrutamiento Explícito: Se logra adicionando el objeto *EXPLICIT_ROUTE* al mensaje *Path*. En el objeto *EXPLICIT_ROUTE* se adicionan todos los nodos por lo cuales se desea que el flujo de tráfico pase.



Observar Ruta Atravesada: se logra adicionando el objeto *EXPLICIT_ROUTE* a los respectivos mensajes *Path* o *Resv*. Dentro de estos mensajes existe una bandera opcional la cual debe ser activada para que se permita almacenar todos los nodos por los que un flujo de datos pasa.

Atribución de Recursos: el proceso de atribución de recursos es implementado por dos prioridades que se encuentran en el objeto *SESSION_ATTRIBUTE* el *Holding Priority* y el *Setup priority*. El *Holding Priority* es la prioridad para retener recursos. El *Setup priority* es la prioridad para tomar recursos. Ambos están en el rango de 0 a 7 donde 0 es la prioridad más alta. *Holding Priority* es la prioridad en la que los recursos asignados en una sesión serán reservados. En una sesión dada, El *Setup Priority* nunca debe ser más alto que el *Holding Priority*.

Cuando una nueva petición de ancho de banda se hace, se comprueba si la petición se puede atender. Si el ancho de banda requerido está disponible pero está en uso por una sesión de prioridad más baja, entonces la sesión de prioridad más baja se puede liberar para que se asigne el ancho de banda necesario a la nueva sesión.

RSVP es un protocolo de señalización y por tanto cumple funciones dentro del plano de reenvío, lo que imposibilita tener una visión general de la topología y de las características de la red MPLS, esto hace que utilizando únicamente este protocolo sea imposible la implementación de algún mecanismo de balanceo de carga. Esto plantea la posibilidad de ser utilizado junto con un protocolo en el plano de control que le suministre la información de la red que le es desconocida, como lo es el ancho de banda libre y utilizado.



CAPÍTULO IV. ANÁLISIS DEL PROTOCOLO OSPF-TE

Este capítulo tiene como objetivo analizar las extensiones realizadas a OSPF para soportar Ingeniería de Tráfico especialmente en redes MPLS. No pretende ser una referencia bibliográfica del protocolo OSPFv2, para tal fin se recomienda consultar directamente el RFC 2328. El capítulo está organizado de la siguiente manera: primero se abordan las generalidades de OSPF y su modo de operación para distribuir la información de enrutamiento, segundo, se analiza el RFC 2370, el cual hace referencia a las mejoras realizadas al protocolo OSPF para soportar una nueva clase de Anuncio de Estado de Enlace (LSA: Link-State-Advertisement) llamados LSAs Opacos. Finalmente, se analizan las extensiones realizadas al protocolo OSPF en el RFC 3630, también llamadas extensiones TE.

4.1 OSPFv2

Es un protocolo de enrutamiento muy utilizado en redes IP. Fue diseñado por el IETF como un protocolo de pasarela interior en reemplazo del protocolo RIP. El protocolo OSPF propone el uso de rutas más cortas y accesibles, mediante la construcción de un mapa de la red, así, es capaz de calcular la métrica para cada ruta y elegir las rutas más cortas. En este proceso se calculan tanto las métricas de estado del enlace (las cuales consisten en asignar pesos de acuerdo al ancho de banda asociado a cada enlace) como de distancia, en el caso de RIP se calcula sólo la distancia y no se tiene en cuenta el tráfico del enlace. OSPF es un protocolo de enrutamiento diseñado para redes con crecimiento constante y capaz de manejar una tabla de enrutamiento distribuida y de rápida propagación [29].

4.1.1 Ventajas

1. Es un protocolo Link State (Estado de Enlace).
2. Rápida detección de cambios en la topología y restablecimiento muy rápido de rutas.
3. Permite la creación de sistemas jerárquicos, utilizando el área 0 como nivel de más



alta jerarquía.

4. División de tráfico por varias rutas equivalentes.
5. Enrutamiento según el tipo de servicio.
6. Soporta Longitud de Máscara de Subred Variable (VLSM: Variable Length Subnet Masks).
7. Utiliza multicasting dentro de las áreas, en redes LAN.
8. Utiliza la división por áreas para crear segmentos lógicos y reducir el tamaño de las tablas de enrutamiento.
9. Autentificación.
10. Luego de la inicialización, OSPF envía únicamente las actualizaciones en las tablas de enrutamiento y no envía la tabla completa como lo hace RIP.
11. OSPF es un estándar abierto y no está relacionado a ningún fabricante en especial [30] [31].

4.1.2 Desventajas

1. Tiene un consumo elevado de CPU.
2. Mantiene múltiples copias de información de enrutamiento, lo cual incrementa la cantidad de memoria requerida.
3. Es un protocolo complejo comparado con otros [30] [31].

4.1.3 Modo de Operación

El protocolo OSPF reconoce tres tipos de conexiones y redes, aunque las extensiones realizadas en el RFC 3036 únicamente abordan el primer tipo de conexión.

1. Líneas punto a punto entre dos dispositivos de enrutamiento.
2. Redes multiacceso con difusión (por ejemplo, la mayoría de redes LAN).
3. Redes multiacceso sin difusión (por ejemplo, la mayoría de redes WAN de conmutación de paquetes).

Debe entenderse por una red multiacceso aquella en la cual varios dispositivos de enrutamiento se pueden comunicar entre sí [31].



4.1.3.1 El Algoritmo Link-State

OSPF es un protocolo link-state. El estado de un enlace es una descripción de la interfaz y de su conexión a enrutadores vecinos. Una descripción de la interfaz podría incluir por ejemplo, la dirección IP, la máscara de subred, el tipo de red conectada, los enrutadores conectados a la red y así, la colección de “link-states” construyen la base de datos del estado de los enlaces.

El protocolo OSPF utiliza un algoritmo link-state en orden a construir y calcular el camino más corto a todos los destinos conocidos. Seguidamente se presentará un breve resumen del funcionamiento del algoritmo de estado del enlace.

1. Luego que un enrutador es inicializado o debido a algún cambio en la información de enrutamiento, el enrutador puede generar un LSA, el cual representa una colección de los estados de enlaces de ese enrutador en particular.
2. Todos los enrutadores pueden intercambiar el estado de sus enlaces por medio de una difusión de LSAs. Cada enrutador que recibe un “link-state” de actualización podría almacenar una copia en su base de datos de estados y luego propagar la actualización a otros enrutadores.
3. Luego que la base de datos de cada enrutador esta completa, podrá ser utilizada para calcular un árbol de caminos más cortos para todos los destinos. OSPF utiliza el algoritmo de Dijkstra para calcular la ruta más corta a un destino. Los destinos, el costo asociado a un enlace, y el próximo salto para alcázar el destino formaran la tabla de enrutamiento.
4. En caso que no se produzcan cambios en la red OSPF, tales como variación en el costo de un enlace, la adición o eliminación de una red, OSPF permanecería en calma. Cualquier cambio que ocurra será difundido y el algoritmo de Dijkstra se activa para recalculer nuevamente el camino más corto [31].

4.1.3.2 LSAs (Link State Advertisements)

En un Sistema Autónomo cada enrutador origina uno o más LSAs y la colección de LSAs conforman la base de datos. El RFC 2328 define los siguientes tipos de LSAs y cada uno de ellos tiene una función específica. Los *Router LSAs* y *Network LSAs*, describen como



los enrutadores y las redes de un área OSPF están interconectadas. Los Summary LSAs proveen una manera de resumir la información de enrutamiento de un área OSPF. Los AS-external-LSAs proveen una manera transparente de anunciar la información externa de enrutamiento, ya sean rutas estáticas o rutas que un enrutador puede recibir de otros protocolos de enrutamiento y hacer su difusión a través del Sistema Autónomo [32].

4.1.3.3 Base de Datos de OSPF

Cada enrutador tiene una base de datos de los estados de los enlaces para cada área a la cual pertenece. Todos los enrutadores pertenecientes a la misma área OSPF tienen idéntica base de datos para dicha área. El algoritmo Shortest Path se calcula de forma separada para cada área [32], esta es una característica favorable para la implementación de mecanismos de Ingeniería de Tráfico con OSPF, dado que los tiempos de convergencia serán tan pequeños como sea la organización por áreas en un Sistema Autónomo.

La base de datos de un área OSPF está compuesta de *Router-LSAs*, *Network-LSAs* y *Summary-LSAs* y con algunas restricciones los *AS-external-LSAs* [32].

4.1.3.4 LSAs-Opacos

Los LSPs Opacos son una clase de LSAs especiales, diseñados para realizar difusión de información de aplicaciones externas a OSPF y aunque utilizan el proceso estándar de difusión de LSAs, no tienen ninguna relación con la información de enrutamiento transportada por los LSAs convencionales.

Los LSAs Opacos son de tipo 9, 10 y 11. Consisten de un encabezado estándar seguido por 32-bits de campo de información específica. La difusión de los LSAs Opacos deben seguir unas reglas de alcance específicas para cada tipo.

- El LSA tipo 9 denota un alcance local. Este tipo de LSA no puede ser difundido más allá de una red o subred local.
- El LSA tipo 10 denota un alcance de área local. Este LSA no puede ser difundido más allá de los bordes del área a la cual está asociado.



- EL LSA tipo 11 denota una difusión en todo el Sistema Autónomo. [33]

4.2 OSPF-TE

El RFC 3630 describe las extensiones TE realizadas al protocolo de pasarela interior OSPFv2 [32], estas extensiones pretenden dar respuesta a los requerimientos planteados en el RFC 2702 (Requerimientos para Ingeniería de Tráfico en MPLS). La semántica de las extensiones de Ingeniería de Tráfico es similar a las de IS-IS. Las extensiones proveen una manera de describir la topología TE (incluyendo ancho de banda y restricciones administrativas) y distribuir la información dentro de una área OSPF.

El RFC 3630 no describe ningún mecanismo para ser utilizado en Ingeniería de Tráfico entre diferentes áreas, dicho tema se plantea como trabajo futuro. Más aun no se plantean cambios en la operación y funcionamiento de la difusión de LSAs en OSPFv2. Cuando existan nodos calificados como No-TE dentro de una topología TE, ellos debe divulgar mensajes LSAs-TE como cualquier otro LSA opaco de tipo 10 [34].

4.2.1 LSA-TE

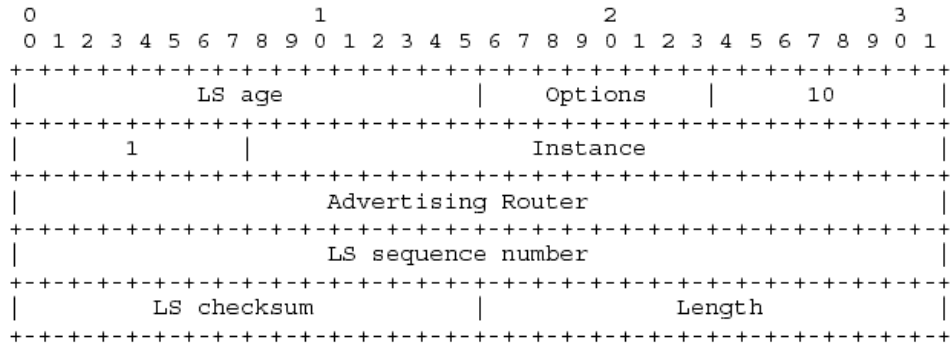
El RFC 3630 define un nuevo LSA Opaco, el LSA-TE. Describe enrutadores y enlaces punto a punto. Para los propósitos de Ingeniería de Tráfico, dado que las extensiones TE únicamente aplican a enlaces punto a punto, los “*Network-LSA*” existentes son suficientes para describir enlaces multi-acceso [34].

Todos los LSAs están identificados por un ID que consta de 8 bits para el tipo de dato y 24 bits para tipo de datos específico. Los LSA-TE utilizan tipo de dato 1, los restantes 24 bits pueden tener un valor arbitrario utilizado para mantener múltiples LSAs de Ingeniería de Tráfico. El ID de un LSA no tiene un significado topológico.

Los LSAs-TE inician con un encabezado estándar, como se muestra en la figura 8 [34].



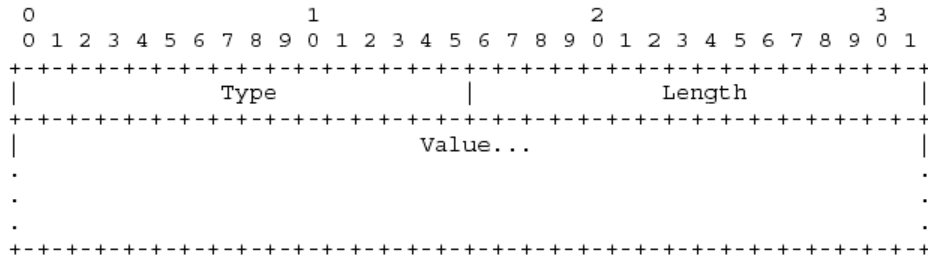
Figura 8: Encabezado de un LSA [34]



4.2.1.1 Carga Útil de un LSA-TE

La carga útil de los LSAs-TE consiste de una o más tripletas Tipo/Tamaño/Valor (TLV: Type Length Value) anidadas, el formato de cada TLV se muestra en la figura 9.

Figura 9: Formato de un TLV [34]



El campo "Length" define el tamaño en bytes de la porción del valor (así un TLV con ningún valor podría tener una tamaño de cero). EL TLV esta alineado a 4 bytes. El relleno no es incluido en el campo de tamaño "Length" (así tres bytes tendrían un tamaño de tres bytes, pero el tamaño total de el TLV podría ser de 8 bytes) [34].

4.2.1.2 Detalles de la carga Útil de un LSA-TE.

Los LSAs-TE contienen TLVs para transportar información, se han definido dos tipos de TLVs: [33]



- Router Address.
- Link.

4.2.1.2.1 TLV "Router Address"

Este TLV especifica una dirección IP estable del enrutador anunciado, la cual siempre es alcanzable si hay conectividad con dicho enrutador. EL atributo clave es la dirección, haciéndose inutilizable si la interfaz esta caída. EL TLV "Router Address" es de tipo 1 y debe aparecer exactamente una vez en el LSAs TE originado por un enrutador [34].

<

4.2.1.2.2 TLV "Link"

Describe un enlace simple y está constituido de varios sub-TLVs, no hay un ordenamiento requerido para los sub-TLVs. Únicamente un "TLV Link" será llevado en cada LSA-TE, permitiendo un fino nivel de granularidad en los cambios topológicos. El "TLV Link" es de tipo 2, y es de tamaño variable.

Los siguientes sub-TLVs de "TLV Link" fueron definidos: [34]

Link type. Es de tipo 1 y define el tipo de enlace que puede ser:

- 1 - Punto-a-Punto.
- 2 - Multi-access.

Link ID. Es de tipo 2 e identifica el otro extremo del enlace. Para enlaces punto-a-punto, es el *Router ID* del vecino. Para enlaces multi-accesos, es la dirección de la interfaz del enrutador designado.

Local interface IP address. Es de tipo 3 y especifica la dirección IP de la interfaz correspondiente a este enlace. Si hay múltiples direcciones locales en un enlace, serán listadas todas en el sub-TLV. Tiene 4N bytes de tamaño, donde N es el número de las direcciones IP locales.

Remote interface IP address. Especifica la dirección IP de las interfaces de los vecinos correspondientes a este enlace. Este atributo y la dirección local son utilizadas para



discernir los múltiples enlaces paralelos entre sistemas. Si el "Link Type" del enlace es de acceso múltiple, la dirección IP remota se fija a 0.0.0.0; Alternativamente, una implementación puede elegir no enviar este sub-TLV. Es de tipo 4 y tiene 4N bytes de tamaño, donde N es el número de direcciones del vecino.

Traffic engineering metric. Especifica la métrica del enlace para propósitos de Ingeniería de Tráfico. Esta métrica puede ser diferente que la métrica estándar de los enlaces en OSPF. Típicamente, esta métrica es asignada por el administrador de la red. Es de tipo 5 y tiene un tamaño de 4 bytes.

Maximum bandwidth. Especifica el ancho de banda máximo que puede ser utilizado en éste enlace, el LSA es originado desde el sistema hacia su vecino. Dice la verdadera capacidad del enlace. Las unidades son bytes por segundo. Es de tipo 6 y tiene un tamaño de cuatro bytes.

Maximum reservable bandwidth. Es de tipo 7 y especifica el máximo ancho de banda que puede ser reservado en este enlace. Notar que este puede ser mayor que el ancho de banda Máximo, en este caso el enlace sería sobre utilizado. Éste es configurado por el administrador; el valor por defecto podría ser el Ancho de banda Máximo. Las unidades están dadas en bytes por segundos y tiene un tamaño de 4 bytes.

Unreserved bandwidth. Es de tipo 8 y especifica la cantidad de ancho de banda que aún no se ha reservado en cada uno de los 8 niveles de prioridad. Los valores correspondientes al ancho de banda pueden ser reservados con un nivel de prioridad de 0 a 7, con orden ascendente, siendo 0 el nivel de más baja prioridad y 7 el nivel de más alta prioridad. Inicialmente todo el ancho de banda estará disponible para ser reservado y cada reserva será menor o igual al máximo ancho de banda reservable. Las unidades están bytes por segundos y su tamaño es de 32 bytes.

Administrative group o Resource Class/Color. Es de tipo 9 y contiene 4 octetos asignados por el administrador de la red. Cada bit corresponde a un grupo administrativo asignado a la interfaz. Un enlace puede pertenecer a múltiples grupos. Por convención, el bit menos significativo es referido como "grupo 0", y el bit más significativo es referido como "grupo 31".



4.2.2 Elementos de procedimiento

Los enrutadores originaran LSAs TE cuantas veces se produzcan cambios en los atributos TE y cuantas veces lo requiera OSPF. Esto no significa que cada vez que se produzca un cambio deba ser difundido inmediatamente, una implementación puede fijar umbrales, (Por ejemplo, un cambio en el umbral del ancho de banda), que activen inmediatamente la difusión e inicie la distribución de otros LSAs luego de un corto periodo de tiempo.

Una vez recibido un cambio de un LSA-TE o LSA de red (Desde que se estén utilizado en cálculos TE), el enrutador podría actualizar su base de datos TE, ni el camino más corto u otros cálculos de rutas son necesarios [34].

4.2.3 Aplicabilidad

El propósito de las extensiones de Ingeniería de Tráfico realizadas al protocolo OSPF es adicionar más atributos a los enlaces, para que sean difundidos por mensajes LSAs dentro de un área OSPF y así poner a disposición de OSPF o una aplicación externa que utilice toda la información que requiera del estado de la red, ya sea para monitorear y hacer un seguimiento del comportamiento de la red o para implementar algún mecanismo de Ingeniería de Tráfico.

La información disponible por las extensiones TE, es utilizada para construir una Base de Datos Extendida del Estado de los Enlaces (ELSDB: Extended Link State Data Base), tal como lo hacen los enrutadores corriendo OSPF para construir la base de datos "regular" del estado de los enlaces (LSDB: Link State Data Base); La diferencia es que la base de datos extendida o base de datos TE, tiene atributos adicionales sobre los enlaces e Incluye:

- Monitoreo de los atributos extendidos de los enlaces.
- Enrutamiento por fuente basado en restricciones.
- Ingeniería de Tráfico Global.

Monitoreo de los atributos: Un dispositivo corriendo OSPF-TE, puede hacer parte de una área OSPF, construir una base de datos TE y consecuentemente reportar el estados de



los atributos TE asociados a cada enlace dentro de un área específica.

Enrutamiento por fuente basado en restricciones locales: Un enrutador puede calcular la ruta desde un nodo A hasta un nodo B. Dicha ruta puede estar sometida a varias restricciones en los atributos de los enlaces y nodos por los cuales atraviesa la ruta. El RFC 3630 no aborda con profundidad el tema de cómo determinar la porción de tráfico que debe ser asignado por las diversas rutas ni tampoco cómo determinar dichas rutas. Entorno a ello únicamente plantea que para determinar la porción de tráfico se utilice la dirección IP destino de los paquetes y para la elección de la ruta, hacer uso de los túneles MPLS.

Ingeniería de Tráfico Global: Un dispositivo puede construir una base de datos TE, una matriz del tráfico de entrada, una función de optimización, que procese la información y calcule de manera óptima o aproximadamente óptima el enrutamiento en toda la red. El dispositivo podría además monitorear la topología TE y reaccionar ante cambios en la red, por ejemplo recalculando las rutas óptimas [34].

4.2.4 Limitaciones

El RFC 3036 únicamente especifica las extensiones y procedimientos para la distribución de información de TE intra-Áreas y las discusiones de cómo podría distribuirse información entre sistemas autónomos se plantea para futuros trabajos.

Las extensiones especificadas únicamente abordan enlaces punto a punto y clarifican que para enlaces con múltiples accesos pueden no ser reflejadas de forma precisa. Tampoco se abordan enlaces sin numeración [34].

4.3 OSPF-API

El OSPF-API [35] permite que aplicaciones externas puedan acceder a la base de datos de OSPF (LSDB), actualmente existe una implementación del OSPF-API basada en el código de Quagga y provee las siguientes funcionalidades:

- Recuperación total o parcial de la LSDB del demonio OSPF. Esto permite que



aplicaciones externas puedan acceder a una copia exacta de la base de datos, incluyendo *Router-LSAs* y *Network-LSAs*. En cualquier momento en que un nuevo LSA llega, el API inmediatamente informa a la aplicación externa enviando un mensaje. Esto permite que siempre se mantenga sincronizada la aplicación externa con la LSDB.

- Permite originar LSA Opacos (9,10,11), los cuales son distribuidos transparentemente a otros enrutadores y recibidos por otras aplicaciones a través del mismo API-OSPF.

4.4 Conclusiones

OSPF-TE permite la distribución de información de estado los enlaces, como lo es BW-Total, BW-Libre, BW-Usado, Métrica de los enlaces e incluso puede transportar información de aplicaciones externas para diferentes fines; todo esto dentro de un área OSPF.

OSPF-TE no garantiza calidad de servicio, puesto que su función es calcular las rutas a los posibles destinos, para tal fin se debe hacer uso de protocolos de señalización que se encarguen de esta tarea.

Es posible la implementación de un mecanismo global de Ingeniería de Tráfico puesto que la información del estado de los enlaces en cada enrutador al interior de un área OSPF es la “misma”.

OSPF puede ser utilizado para implementar sistemas de monitoreo en la red, dado que la distribución de información mediante los LSAs Opacos es de forma transparente y puede ser extraída en cualquier nodo de la red.

Existe gran interés en la investigación de las extensiones para Ingeniería de Tráfico, el cual ha llevado al desarrollo de un OSPF-API, que habilita aplicaciones externas para interactuar directamente con la base de datos de OSPF.



CAPÍTULO V. BALANCEO DE CARGA EN REDES MPLS UTILIZANDO LOS PROTOCOLOS RSVP-TE Y OSPF-TE

Este capítulo tiene como objetivo determinar las características deseables que deberá tener el mecanismo de balanceo de carga a implementar, teniendo en cuenta algunas recomendaciones de los RFCs 2702 [1] y 3272 [10], como también determinar cuáles de los mecanismos de balanceo de carga analizados en el capítulo 2 permitirían obtener dichas características, y así seleccionar el mecanismo de balanceo de carga que más se aproxime a las características deseadas. Finalmente evaluar si es viable la implementación del mecanismo de balanceo de carga seleccionado haciendo uso de las extensiones para ingeniería de tráfico que proveen los protocolos RSVP-TE [28] y OSPF-TE [34].

5.1 CARACTERÍSTICAS DESEABLES QUE EL MECANISMO DE BALANCEO DE CARGA DEBE PRESENTAR

El mecanismo de balanceo de carga que se propone implementar esta dirigido a operar en el backbone de la red, puesto que este es el principal campo de operación de la tecnología MPLS. Consecuentes con el objetivo de balanceo de carga planteado en el capítulo 2, este mecanismo tendrá como objetivo principal establecer un control dinámico de los posibles estados de congestión que presenten las redes MPLS, y en adelante será referenciado como Balanceo de Carga Dinámico con Control de Congestión en redes MPLS (DLCCM: Dynamic Load-Balancing with Congestion Control in MPLS).

5.1.1 Tipo de Control

El control que debe implementar el mecanismo DLCCM debe ser auto-controlable [1], es decir debe ajustarse dinámicamente a los cambios que presente la red, por esta razón se ha elegido el tipo de control Reactivo [10], ya que este tipo de control es utilizado en algoritmos de baja complejidad, lo que permite trabajar en equipos de red que por lo general poseen limitaciones tanto de memoria como de CPU, y se ajusta a los cambios en la distribución del tráfico presente en los enlaces de Internet.



Puesto que el balanceo de carga es una de las funciones de control de enrutamiento que está en la escala de tiempo de *término medio*, y va desde milisegundos a días, se debe garantizar que el mecanismo DLCCM opere en este rango de tiempo [10].

El mecanismo DLCCM debe ejecutar el control desde la *Oferta de Recursos* [10], ya que esto permite incrementar la capacidad efectiva de ancho de banda disponible en la red. El *Control de la Oferta* también permite alcanzar el objetivo principal del mecanismo DLCCM, el cual consiste en realizar balanceo de carga para reducir los niveles de congestión en las redes MPLS.

5.1.2 Tipo de Medidas

Debido a que el mecanismo DLCCM tendrá un tipo de control reactivo y actúa sobre la oferta de recursos, las medidas se deben realizar en línea, de tal manera que la información obtenida sea una representación aproximada del estado actual de la red y se garantice que los tiempos de actualización estén en la escala *temporal media*, establecida para el tipo de control reactivo [10].

5.1.3 Caracterización

El principal parámetro que debe evaluar el mecanismo DLCCM es el ancho de banda, puesto que es el recurso que se debe preservar, a fin de reducir la congestión, minimizar los retardos medios, disminuir las pérdidas de paquetes y de esta manera mejorar el desempeño de la red. Este parámetro puede ser monitoreado mediante medidas del máximo ancho de banda, ancho de banda utilizado y ancho de banda libre asociado a cada uno de los enlaces de la red.

Para evaluar el desempeño del mecanismo DLCCM, se deben realizar medidas del ancho de banda efectivo transportado sobre la red, y así verificar que el sistema opere dentro de los umbrales establecidos administrativamente, como también realizar medidas de los tiempos de estabilización, correspondientes al tiempo que tarda el mecanismo DLCCM en eliminar o reducir un posible estado de congestión ya presente en la red, y así determinar el buen funcionamiento del mecanismo DLCCM en un dominio MPLS.



5.1.4 Integración de los procesos de enrutamiento con los de conmutación

Para lograr una integración entre los procesos de enrutamiento y los procesos de conmutación se hará uso de la tecnología MPLS, a fin de unificar las capacidades y potencialidades del plano de control con la rapidez del plano de conmutación, de tal manera que los procesos realizados en un plano sean reflejados en el otro [1].

5.1.5 Todos los anteriores a un Bajo Costo

El mecanismo DLCCM debe hacer uso de los recursos disponibles en la red, de tal manera que los gastos de implementación sean moderados. Esto se obtiene en gran medida utilizando tecnologías estándares, haciendo que las actualizaciones de software y hardware sean mínimas.

5.2 ANÁLISIS FUNCIONAL DE LOS MECANISMOS DE BALANCEO DE CARGA EXISTENTES EN REDES MPLS

En este apartado se presenta un análisis funcional de los mecanismos de balanceo de carga presentados en el capítulo 2, con el objetivo de determinar si sus características funcionales coinciden con las características deseadas en el mecanismo DLCCM que se pretende implementar y a la vez determinar la viabilidad de implementación, haciendo uso de los protocolos RSVP-TE y OSPF-TE.

5.2.1 Análisis Funcional de los mecanismos CBR

Los mecanismos CBR utilizan generalmente el tipo de control pro-activo, a fin de predecir y evitar estados indeseables de la red, por lo que utilizan algoritmos complejos que en muchas ocasiones consumen elevados niveles de procesamiento, haciendo poco viable su implementación.

Las medidas realizadas por estos mecanismos son generalmente fuera de línea, lo cual les impide obtener las características y el estado actual de la red, impidiendo así detectar algunos problemas en la red como lo son los cuellos de botella.



Los mecanismos CBR son una mejora de los protocolos de enrutamiento convencionales, por lo cual operan solo en el plano de control, lo que les impide tener una visión general de la red física, dando como resultado que en algunas ocasiones, aunque las decisiones a nivel de control sean óptimas tengan consecuencias indeseables en el plano de conmutación.

La implementación de los mecanismos CBR es costosa, debido a su alto nivel de complejidad que en ocasiones requieren grandes actualizaciones de Hardware para poder ser implementados.

Acorde a lo expuesto anteriormente se puede determinar que este tipo de mecanismos presentan características que difieren de las deseadas en el mecanismo DLCCM, por esta razón no serán tenidos en cuenta para una posible implementación utilizando los protocolos RSVP-TE y OSPF-TE.

5.2.2 Análisis Funcional de los Mecanismos Divisores de Tráfico

Los mecanismos divisores de tráfico analizados en el capítulo 2 presentan grandes ventajas respecto a los mecanismos tipo CBR, una de ellas radica en el tipo de medidas que realizan para conocer el estado actual de la red, dichas medidas son realizadas en línea, lo que les permite monitorear parámetros relevantes en el desempeño de la red, como lo son la cantidad paquetes perdidos, retardos medios, etc. Por ello en este apartado se realizará un análisis funcional de cada uno de estos mecanismos divisores de tráfico y se determinará la viabilidad de implementación haciendo uso de los protocolos RSVP-TE y OSPF-TE.

5.2.2.1 Análisis Funcional de MATE

MATE es un algoritmo potente el cual utiliza un tipo de control reactivo y su principal objetivo es reducir los niveles de congestión en la red, reaccionando ante las medidas en línea realizadas sobre la misma. Aunque el tipo de medidas que realiza es en línea, el parámetro y la forma en que hace las medidas le introduce un gran porcentaje de error al momento de hacer la división de tráfico. Las medidas se realizan mediante un paquete de prueba que es enviado sobre la red, a partir de este paquete se miden parámetros como



lo son los retardos medios y cantidad de paquetes perdidos, con lo cual se extrae la información que determinaría el estado actual de la red. Teniendo en cuenta que en MPLS los caminos que se establecen son unidireccionales, se puede dar el caso que los paquetes de prueba tengan un camino de retorno diferente al camino por el que fueron introducidos en la red, otra falla esta en que la perdida de paquetes puede tener diversas causas y no solamente la sobrecarga de un enlace.

MATE aborda un aspecto importante del tráfico en las redes, como lo es la clasificación del tráfico que requiere ser balanceado y el tráfico que se incorpora en los nodos intermedios o tráfico cruzado, que podría saturar un enlace y que mediante las medidas realizadas en la red podría ser detectado.

Acorde a los requerimientos de MATE, podría ser implementado haciendo uso de RSVP-TE y OSPF-TE. RSVP-TE sería utilizado para la creación de los LSPs extremos a extremo, en tanto que OSPF-TE únicamente sería utilizado para funciones del plano de control convencionales. Sin embargo su diferencia principal con el mecanismo DLCCM propuesto esta en la caracterización, puesto que no tiene en cuenta el ancho de banda efectivo en la red, siendo este el parámetro principal que debe evaluar el mecanismo DLCCM.

5.2.2.2 Análisis Funcional de FATE

FATE es un mecanismo que al igual que MATE tiene sus grandes deficiencias al momento de realizar las medidas, debido a que los parámetros que evalúa para determinar un posible estado de congestión en la red son la cantidad de paquetes perdidos o los requerimientos fallidos en la creación de un nuevo LSP, los cuales impiden tener información precisa del estado real de la red. Una vez detectado el estado de congestión, FATE entraría a realizar el balanceo de carga, el cual consiste en reenrutar LSPs que estén congestionando un enlace. Como se puede notar esta técnica no es muy recomendada cuando se tienen un número reducido de LSPs por enlace, puesto que al reenrutar LSPs con gran cantidad de ancho de banda, fácilmente se podría dejar al borde de la saturación el enlace al cual se ha balanceado la carga y esto podría causar el efecto de ping-pong, introduciendo inestabilidad en la red.



FATE podría ser implementado haciendo uso de los protocolos RSVP-TE y OSPF-TE, pero igual que MATE tiene su principal falla en la caracterización de los parámetros que evalúa para determinar el estado de la red, dejando sin monitorear el ancho de banda efectivo en la red.

5.2.2.3 Análisis Funcional de DLB

Este mecanismo utiliza un control de tipo reactivo, aunque no plantea cómo se deben hacer las medidas, en la caracterización define como parámetro principal el ancho de banda efectivo de la red, por tanto se puede deducir que las medidas deben ser realizadas en línea.

DLB podría ser implementado haciendo uso de RSVP-TE para crear los LSPs explícitos extremo a extremo, sobre los cuales se realizaría el balanceo de carga y OSPF-TE para distribuir el estado de los enlaces sobre toda la red. Como se puede notar se debe hacer uso de un mecanismo externo que recolecte la información y la entregue a OSPF-TE para que se encargue de hacer la distribución. Su principal inconveniente parte del hecho de conocer la cantidad de ancho de banda de cada nuevo flujo, lo cual impone una gran restricción, haciendo que su implementación solo pueda ser realizada en una red donde la asignación de recursos sea bajo reserva.

5.2.2.4 Análisis Funcional de Delay-based Adaptive Load Balancing Algorithm

Es un mecanismo de control reactivo y similar a MATE tiene en cuenta el tráfico a balancear y el tráfico que es introducido en los nodos intermedios. Las medidas realizadas son en línea, sin embargo no tiene en cuenta en la caracterización el ancho de banda efectivo en la red.

En su implementación podría hacer uso de RSVP-TE para crear los caminos explícitos extremo a extremo y OSPF-TE como protocolo de enrutamiento, sin hacer uso de las extensiones de ingeniería de tráfico.



5.2.2.5 Análisis Funcional de DYLBA

DYLBA es un mecanismo que se aproxima funcionalmente al mecanismo DLCCM. El mecanismo de control que utiliza es de tipo reactivo, las medidas realizadas son en línea, en la caracterización que realiza tiene como parámetro principal el ancho de banda. El problema que presenta se debe a su principio de balanceo de carga, el cual se basa en el re-enrutamiento de LSPs, lo cual no es recomendable si se tiene un número reducido de ellos.

En su implementación puede hacer uso de RSVP-TE para establecer las rutas extremo a extremo, además utiliza las capacidades de re-enrutamiento con las que cuenta este protocolo. En el caso de OSPF-TE tendría una función muy importante, pues haría uso de las extensiones TE para distribuir hasta el borde de la red el estado de los enlaces, es decir el ancho de banda asociado a cada uno de ellos.

5.2.2.6 Análisis Funcional de LCM

LCM es el mecanismo de balanceo de carga que más se aproxima a mecanismo DLCCM. Utiliza un sistema de control reactivo, las medidas realizadas son en línea, el parámetro principal en la caracterización es el ancho de banda, su principio básico de balanceo de carga esta en dejar intacta la configuración de los LSPs y mover el tráfico entre los diversos LSPs configurados, de esta manera no genera inestabilidad en la red.

Su implementación podría ser realizada utilizando RSVP-TE para establecer las rutas explícitas extremo a extremo, en tanto que haría uso de OSPF-TE para funciones de ingeniería de tráfico en el core, consistentes en distribuir la información del estado de los enlaces la cual sería analizada en el borde donde estaría implementado el mecanismo DLCCM.



CAPITULO VI. IMPLEMENTACIÓN Y VALIDACION DEL MECANISMO DLCCM

El objetivo principal de este capítulo es la implementación y validación del mecanismo de balanceo DLCCM. Como primera parte se presentan las principales funciones utilizadas en la implementación del mecanismo DLCCM, seguidamente se da una explicación detallada del funcionamiento del algoritmo de balanceo que utiliza el mecanismo DLCCM. Finalmente se realizan las pruebas necesarias, a fin de validar su funcionamiento.

6.1 IMPLEMENTACIÓN DEL MECANISMO DLCCM

Para la implementación del mecanismo DLCCM se tendrá en cuenta el análisis realizado en el capítulo V, donde se estudiaron las características funcionales de cada uno de los mecanismos y algoritmos de balanceo de carga presentados en el capítulo II. Se concluye que utilizando el algoritmo LCM como algoritmo de balanceo de carga, junto con las funcionalidades de Ingeniería de Tráfico de los protocolos RSVP-TE y OSPF-TE, se pueden obtener características funcionales similares a las del mecanismo DLCCM que se pretende implementar.

6.1.1 Topología de Red

Antes de implementar el mecanismo de balanceo de carga, es necesario definir la topología de red que se utilizará para validar su funcionamiento. Teniendo en cuenta que los recursos computacionales son escasos, se optó por elegir e implementar una red MPLS experimental, donde existieran mínimo dos caminos con diferentes pesos y así tener la disponibilidad de realizar balanceo de carga entre ellos.

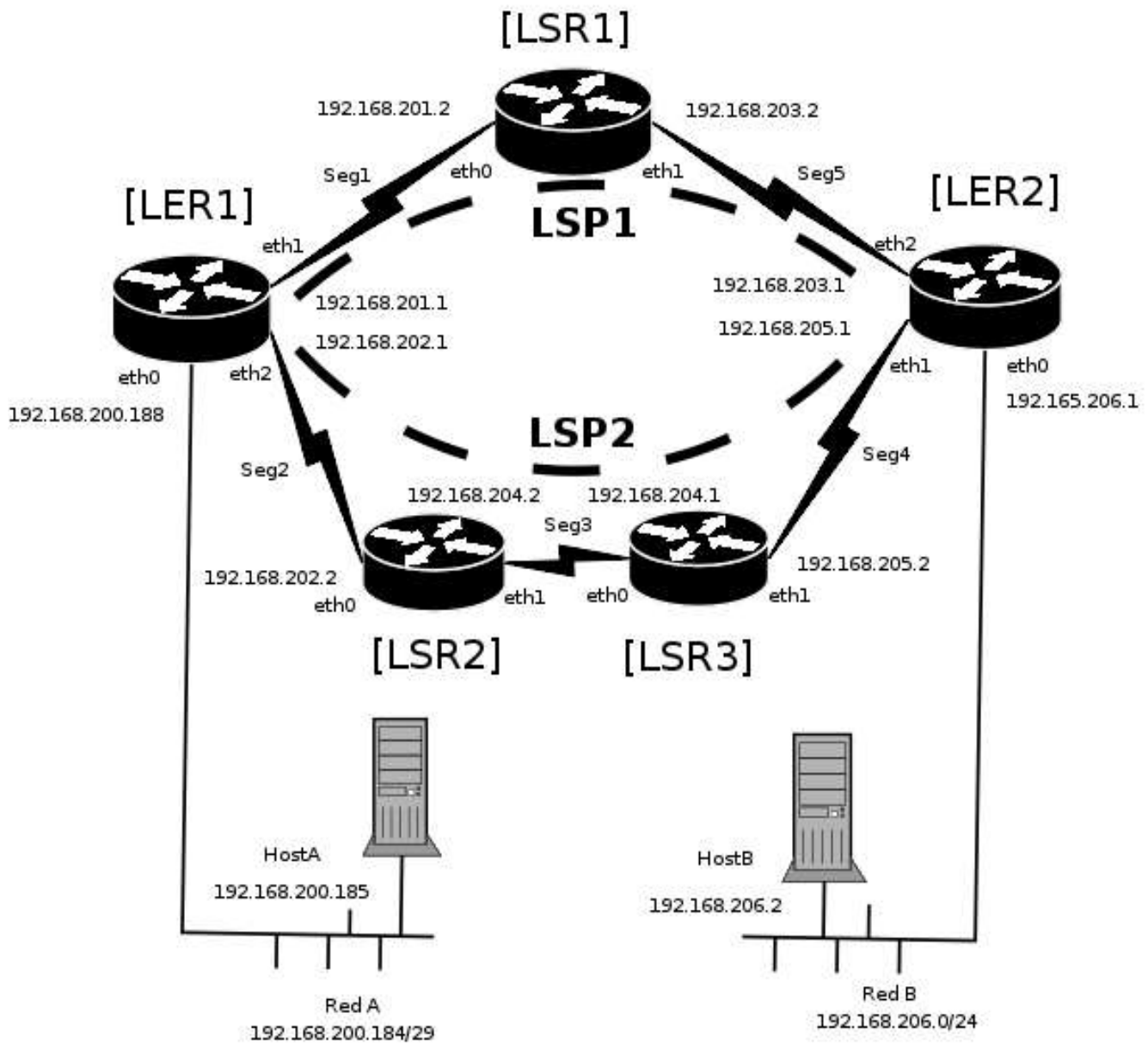
La Figura 10 muestra el diagrama de red seleccionado, la red MPLS experimental está compuesta por un backbone MPLS, conformado por tres LSRs, dos LER uno de ingreso y otro de egreso; también se requirió de dos PC's que representarían los clientes finales. El algoritmo de balanceo será implementado en el LER1, el cual dispondrá de dos LSPs para alcanzar el LER2. El LSP principal (Lp) esta compuesto por LER1-LSR1-LER2 y el



LSP secundario (Ls) esta conformado por LER1-LSR2-LSR3-LER2.

Es importante anotar que todos los equipos que conforman la red MPLS experimental tienen sistema operativo LINUX.

Figura 10: Red MPLS Experimental

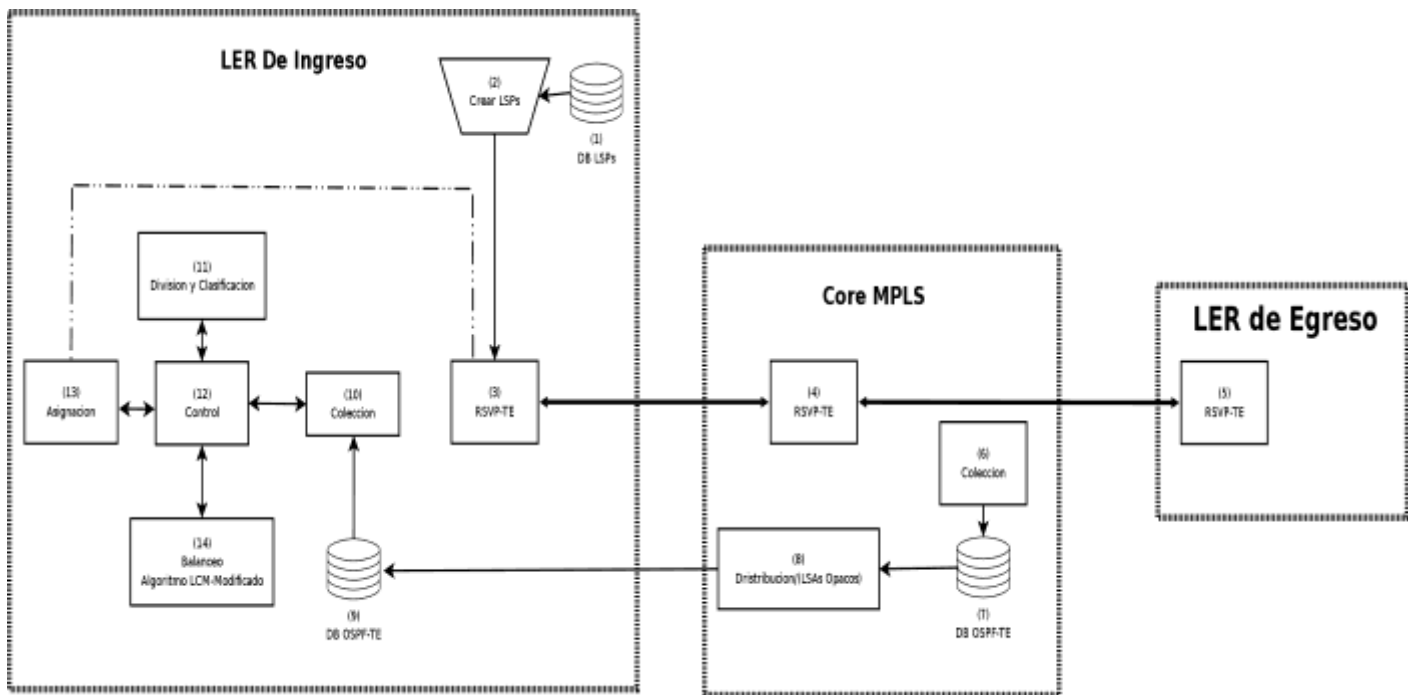


6.1.2 División Funcional del Mecanismo DLCCM

En este apartado se hará una presentación general de las principales funciones que integran el mecanismo DLCCM, en el anexo A se encuentra el código total utilizado para la implementación.

El mecanismo DLCCM esta compuesto por diferentes funciones distribuidas que cumplen una tarea especifica, las funciones principales son: Colección, Distribución, División de Tráfico, función de Asignación, función de Balanceo y Control. El diagrama funcional en bloques se muestra en la figura 11.

Figura 11: Diagrama en Bloques del Mecanismo de Balanceo de carga



Cada una de las funciones componentes del mecanismo DLCCM está implementada en Bash [36]. Aunque bash es un lenguaje de programación interpretado y no tiene la misma potencia y rapidez que C, es suficiente para automatizar el proceso de balanceo de carga, puesto que el tipo de control que se está utilizando es reactivo, lo cual hace que el algoritmo sea de baja complejidad de implementación, además este lenguaje permite



integrar fácilmente los proyectos quagga [37] y rsvpd [38], los cuales implementan algunas de las características de los protocolos OSPF-TE y RSVP-TE respectivamente, así como utilizar fácilmente las herramientas disponibles en Linux, como el caso de iptables [39], para hacer el mapeo de las FECs sobre los LSPs.

6.1.2.1 Función de Colección

La función de colección es la encargada de realizar un cálculo estadístico de la tasa de transmisión de datos en cada uno de los equipos que conforman la red MPLS experimental, concretamente en los LSRs y los LERs se ejecuta un proceso cada 10 segundos que se encarga de calcular la velocidad de datos de cada una de las interfaces de red. Puesto que el entorno de trabajo es LINUX, los datos utilizados para realizar dicho cálculo serán extraídos del archivo “/proc/net/dev” y el resultado del cálculo en los LSRs será almacenado en un archivo de texto llamado “/root/bin/report/opaque_1sa”, específicamente en el sub-TLV “Maximum Reservable Bandwidth” de OSPF-TE y definido por quagga como “mpls-te link max-rsv-bw”. En los LERs de ingreso se almacenará la información local del estado actual de los LSPs, en los archivos “/root/bin/report/T11680eth1” y “/root/bin/report/T11680eth2”. En este caso solo existirán dos archivos, debido a que únicamente se crearán dos LSPs.

6.1.2.2 Función de Distribución

La función de Distribución tiene la tarea de leer el estado actual de los enlaces, información que previamente la función de Colección ha registrado en el archivo “/root/bin/report/opaque_1sa”, e introducir sus datos en la base de datos de OSPF-TE, para que OSPF mediante su proceso estándar de distribución de LSAs Opacos difunda la información en toda la red MPLS experimental.

El software de enrutamiento quagga tiene por defecto deshabilitadas las extensiones para ingeniería de tráfico, por lo tanto se debe instalar desde el código fuente y en la configuración habilitar los LSAs Opacos, las extensiones TE y la interfaz de comandos vtysh, como se muestra a continuación:

```
[root@sarge quagga-0.99.4]# ./configure --enable-opaque-1sa --enable-ospf-te --enable-vtysh
```




Luego de habilitar las extensiones TE y los LSAs Opacos, se debe configurar OSPF para que haga uso de ellas, ejemplo:

```
sarge# conf t  
sarge(config)# router ospf  
sarge(config-router)# capability opaque  
sarge(config-router)# mpls-te on  
sarge(config-router)# mpls-te router-address 192.168.201.2
```

Una vez configurado OSPF para la distribución de información TE, se inicia la distribución de esta información y cambios que puedan presentarse en cada uno de los atributos TE, sobre toda el área OSPF. Acorde a las extensiones para ingeniería de tráfico disponibles en OSPF, se utilizarán los siguientes sub TLVs para distribuir la información de estado de los enlaces:

```
Maximum Bandwidth: 10000 (Bytes/sec)  
Maximum Reservable Bandwidth: 0 (Bytes/sec)  
Resource class/color: 0xa2
```

EL sub TLV “Maximum Bandwidth”, contiene el máximo ancho de banda disponible en el enlace. En la implementación del mecanismo DLCCM el sub TLV “ Maximum Reservable Bandwidth” (definido por el estándar como el atributo que lleva la información del ancho de banda que puede ser reservado en el enlace) se utilizará para transportar la tasa de transferencia de datos actual asociada a cada enlace. El sub TLV “Administrative group o Resource Class/Color”, se utiliza para identificar los LSPs que van desde el LER de ingreso a el LER de egreso, los cuales tienen asignado un valor arbitrario de “0xa2”, los que van en sentido opuesto tienen un valor de “0xa1”; si bien el balanceo de carga solo será implementado en un sentido, fácilmente se podría implementar de forma simultanea desde el sentido opuesto.

Es importante anotar que la función de distribución únicamente tiene sentido en los LSRs, ya que en los LERs la lectura del estado actual de los enlaces se hace localmente.



6.1.2.3 Función de División y Clasificación

Es la encargada de dividir el tráfico que llega al LER de Ingreso, esta división como se menciona en el capítulo 2, se puede hacer por paquetes o por flujos, para la presente implementación se determinó utilizar la división por flujos, puesto que garantiza la secuencia de los paquetes, lo cual constituye una gran ventaja frente a la división por paquetes. Uno de los inconvenientes que presenta la división de tráfico por flujos, se debe al nivel de granularidad que puedan tener los flujos, debido a que si no existe gran cantidad de flujos o el tráfico cursado por cada uno de ellos difiere considerablemente, el balanceo de carga puede llegar a ser muy impreciso. El problema de granularidad en los flujos puede ser solucionado utilizando quintupletas de parámetros para clasificar cada flujo, así el nivel de granularidad sería alto y cada flujo estaría clasificado por su dirección IP origen, dirección IP destino, puerto de origen, puerto de destino y protocolo.

La clasificación del tráfico por flujos, corresponde a crear una FEC por cada flujo de datos, luego cada paquete de datos perteneciente a una FEC, será reenviado por el dominio MPLS bajo las mismas características y restricciones establecidas para la FEC. Dicha clasificación del tráfico por flujos, es posible realizarla en LINUX utilizando el módulo "ip_conntrack" de Iptables, el cual registra cada una de las conexiones establecidas en la máquina, sobre en el archivo "/proc/net/ip_conntrack" y cada vez que una conexión se termina también desaparece del archivo, lo cual es de gran utilidad para mantener un registro actualizado de los flujos existentes en la red. Existe un número máximo de conexiones por defecto que pueden ser registradas y está definido en el archivo "/proc/sys/net/ipv4/ip_conntrack_max". El valor máximo por defecto depende de la cantidad de memoria RAM que tenga el PC, pero este valor puede ser modificado mediante la siguiente orden:

```
[root@puffy bin]# echo "16344" > /proc/sys/net/ipv4/ip_conntrack_max
```

Las funciones de división y clasificación están implementadas en la función "dctráfico", localizada en el archivo "lbmplsV01.sh" en el LER1.



6.1.2.4 Función de Asignación

Es la función encargada de asignar una FEC al respectivo LSP por la cual debe ser enviada. Para realizar esta asignación se utilizará la tabla “mangle” de iptables, la cual permite asignar una marca a los paquetes que pertenezcan a una FEC previamente establecida por la función de división, dicha marca es utilizada para hacer enrutamiento en base a marcas, y así determinar sobre cual LSP debe ser enrutada cada FEC. Ejemplo de asignación de una FEC a un LSP:

```
iptables -t mangle -D PREROUTING -s src_ip -d dst_ip -p tcp --sport src_port --dport  
dst_port -j MARK --set-mark ID
```

```
iptables -t mangle -D OUTPUT -s src_ip -d dst_ip -p tcp --sport src_port --dport  
dst_port -j MARK --set-mark ID
```

Como se puede ver en el ejemplo anterior, el objetivo “MARK”, tiene la función de asignar una marca de valor “ID” al paquete que corresponda con el filtro src_ip, dst_ip, tcp, src_port y dst_port. El valor ID corresponde al valor del LSP-ID de un camino previamente configurado.

6.1.2.4 Función de Balanceo

Ejecuta el algoritmo de balanceo de carga LCM-MG (se describe en detalle en la sección 6.2), el cual determina el número de FECs y en que momento deben ser asignadas a un determinado LSP, como también el momento en que deben ser eliminadas.

En esta función se implementa el método de balanceo de carga por flujos, de tal manera que cada que se necesite balancear una determinada cantidad de tráfico, se hará balanceando cierta cantidad de flujos. Esto se hace debido a que se desconoce el tamaño de tráfico por flujo. De esta manera cada vez que se necesite balancear una cantidad de ancho de banda BW, se balanceará un flujo de tráfico y luego se observará en la interfaz local si la cantidad BW ha sido balanceada, si no es así se seguirá balanceando flujo por flujo hasta que se logre balancear la cantidad deseada.

Cada vez que se balancee un flujo de tráfico se adiciona un retardo llamado hold time (ht)



para que la interfaz local tenga tiempo de actualizar los datos. El tiempo ht elegido es de 10sg, ya que fue el menor tiempo que se puede retardar el proceso para que la lectura de la interfaz local se actualice y presente un mínimo de error.

Un determinado ancho de banda BW a balancear esta conformando por flujos de tráfico, la cantidad de flujos será mayor si la cantidad de tráfico por flujo es pequeña.

Debido a la adición del tiempo de retardo “ht”, cuando se requiera balancear un determinado ancho de banda BW, se necesitará más tiempo para el balanceo, si el tráfico por flujo es pequeño, ya que esto quiere decir que el número de flujos a balancear es grande, siendo este proporcional al retardo de balanceo que el algoritmo adiciona.

La función de balanceo está implementada en el archivo “/root/bin/lbmplsV01.sh”, localizado en el LER de Ingreso.

6.1.2.5 Función de Control

Determina el control total del mecanismo de balanceo de carga, el tiempo de lectura de los estados de los enlaces, como también cada cuánto tiempo se estará ejecutando el algoritmo de balanceo de carga, por defecto se ejecuta cada 30 segundos, pero este valor puede ser configurado por el administrador, asignando el tiempo en segundos a la variable “PERIODO”, localizada en la sección de variables globales en el archivo “lbmplsV01.sh”, donde está implementada la función de control, este archivo contiene otras funciones secundarias que aun no han sido mencionadas y que contribuyen al correcto funcionamiento del mecanismo, entre ellas están: la función “readtun”, es un complemento de la función de colección en el LER1, tiene la tarea de obtener el estado actual de cada LSP, por tanto debe acceder a la base de datos de OSPF-TE, extraer la información del estado de las interfaces de los LSRs contenidos en el backbone MPLS y compararlos con el estado actual de las interfaces locales a fin de seleccionar el máximo valor que determina el ancho de banda utilizado en cada LSP. Otras pequeñas subfunciones han sido implementadas con el fin de reducir la complejidad de código de las funciones principales y hacer más fácil la depuración de errores en tiempo de implementación, todas estas funciones se encuentran documentadas en el anexo A.



6.2 ALGORITMO QUE UTILIZA EL MECANISMO DLCCM

En este apartado se presenta un análisis del algoritmo de balanceo de carga utilizado en el mecanismo DLCCM. Primero se muestra el funcionamiento básico del algoritmo LCM, seguidamente se presentan las falencias encontradas en dicho algoritmo. Finalmente se proponen modificaciones al algoritmo LCM para solucionar sus falencias.

6.2.1 Funcionamiento del Algoritmo LCM

El algoritmo LCM original, parte de dos supuestos para que su implementación sea exitosa. El primero, que los LSPs sean previamente configurados y el segundo que la información del ancho de banda asociado a cada LSP sea un parámetro conocido al instante de ejecutar el algoritmo. Por ahora se supondrá que dichos requerimientos se cumplen satisfactoriamente, y más adelante se expondrán los procesos necesarios para obtener dichos requerimientos.

El algoritmo LCM define dos umbrales de utilización llamados umbrales de congestión “C”, y umbral de utilización Medio “M”. Puesto que el algoritmo tiene un sistema de control reactivo, cada vez que detecta que el umbral “C” es superado en un enlace, balancea el tráfico que excede este umbral sobre el siguiente LSP disponible, es decir balancea el exceso de carga por las rutas alternas disponibles. El umbral “M” indica que si la carga sobre un enlace es menor o igual a este umbral y existe tráfico por un enlace cuyo peso es mayor, parte de ese tráfico será retornado a la ruta de menor peso, esto con el fin de dar un nivel de prioridad a las rutas más cortas, sin dejar de utilizar las rutas alternas cuando los requerimientos de ancho de banda no se pueden satisfacer por las rutas principales o rutas con menor peso.

Los umbrales de congestión “C” y de utilización media “M”, son definidos por el administrador, pero en cualquier caso debe cumplirse la siguiente regla:

$$\text{Max} > \text{C} > \text{M}$$

Donde “Max”, es el máximo ancho de banda utilizable en cada LSP, de tal manera que



cuando en un camino los niveles máximos de utilización sean diferentes, siempre se debe tomar el menor valor, el cual puede ser diferente para cada LSP. Los valores “C” y “M” son porcentajes referidos al máximo ancho de banda utilizable (Max) y son definidos administrativamente, ejemplo:

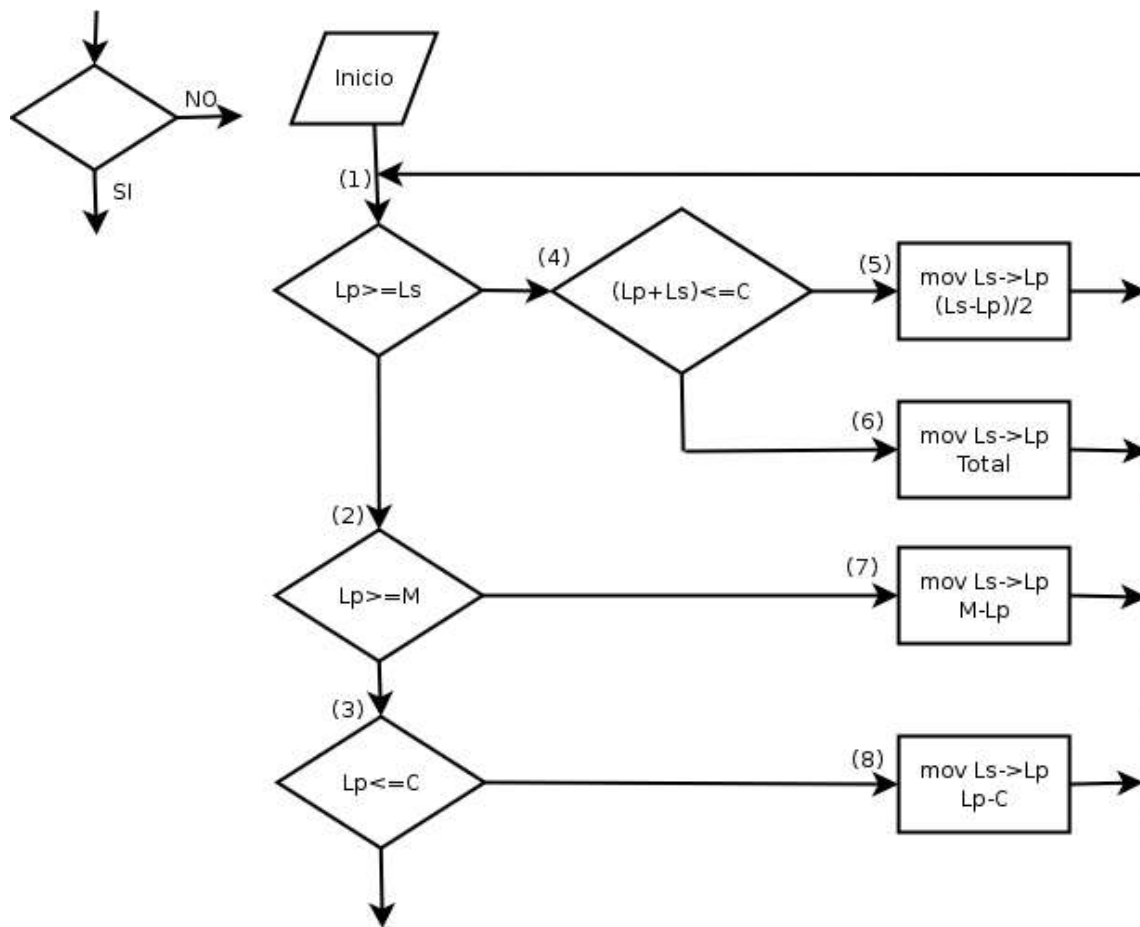
Max=1000Kbps

C=70%Max=700Kbps

M=50%Max=500Kbps

La figura 12 muestra el funcionamiento lógico del algoritmo LCM, mediante el diagrama de flujos respectivo.

Figura 12: Diagrama de Flujos del Algoritmo LCM



Seguidamente se analizará el algoritmo LCM balanceando carga sobre dos LSPs, uno denominado primario o Lp y el otro secundario o Ls.



Inicialmente la carga de tráfico se envía sobre el Lp, el cual se configura como la ruta más corta. El análisis respectivo del algoritmo LCM se representa por los siguientes estados:

(1). Se observa si la carga de tráfico en Lp es mayor o igual que la carga de Ls. Si SI va a (2), si NO va a (4).

(2). Consultar si Lp es mayor o igual a "M", si SI va a (3), si NO va a (7).

(3). Consultar si Lp es menor o igual a "C", si SI, indica que el tráfico de Lp (LSP primario) está entre los umbrales "M" y "C", por lo tanto no debe realizar cambios en la distribución de tráfico, el algoritmo termina y retornará nuevamente a (1), si No va a (8)

(4). Consultar si la suma del tráfico cursado en Lp y Ls es menor o igual que "C", si SI va a (6), si NO va a (5).

(5). Lp tiene ancho banda disponible y la carga del Ls es mayor que la transportada por Lp; sin embargo la totalidad del tráfico en Ls no puede ser transportado por Lp, por tanto se mueve una cantidad de tráfico aproximado de $(Ls-Lp)/2$, esto con el fin de equilibrar la carga entre los enlaces. El algoritmo termina y retorna a (1).

(6). Debido a que todo el tráfico puede ser cursado por Lp sin que exista congestión, se ejecuta un proceso que mueve la totalidad del tráfico que se está cursando por Ls a Lp. El algoritmo termina y retorna a (1).

(7). El tráfico cursado en Lp es menor que el umbral de trabajo "M", se ejecuta un proceso que mueve una cantidad de tráfico $(M-Lp)$ desde Ls a Lp, con el fin de aumentar la utilización del enlace principal. El algoritmo termina y va a (1).

(8). Se ha detectado congestión en el enlace primario Lp, por tanto se ejecuta el proceso de mover una cantidad de tráfico $(Lp-C)$ desde Lp hacia Ls. Con lo cual se elimina el estado de congestión transitorio detectado en la red. El algoritmo termina y va a (1).



Falencias del algoritmo LCM:

Luego de hacer un análisis detallado del funcionamiento del algoritmo LCM, se encontraron las siguientes falencias:

- La primera deficiencia fue detectada cuando se pretendía mover una cantidad de tráfico (M-Lp) desde Ls a Lp, siendo Ls menor que la cantidad de tráfico (M-Lp), ocasionando que el algoritmo LCM entrara en un bucle infinito, y por tanto que el mecanismo DLCCM falle.
- La segunda deficiencia fue detectada cuando el enlace primario estaba en congestión y el segundo entraba a ese estado, de tal manera que cuando ingresaba a la red un nuevo tráfico, el enlace primario balanceaba la carga hacia el enlace secundario, haciendo que el tráfico de este enlace fuera mayor que el del primario, bajo estas condiciones el algoritmo LCM detectaba que $L_s > L_p$ y balanceaba carga del enlace secundario al primario, formando un bucle infinito y haciendo que el tráfico oscilara de un enlace a otro.
- La tercera falla detectada se debe a los problemas de escalabilidad que presenta el algoritmo LCM, debido a que únicamente permite hacer balanceo de carga entre dos LSPs y en una implementación real se pueden tener múltiples LSPs configurados, sobre los cuales se debe hacer balanceo de carga.

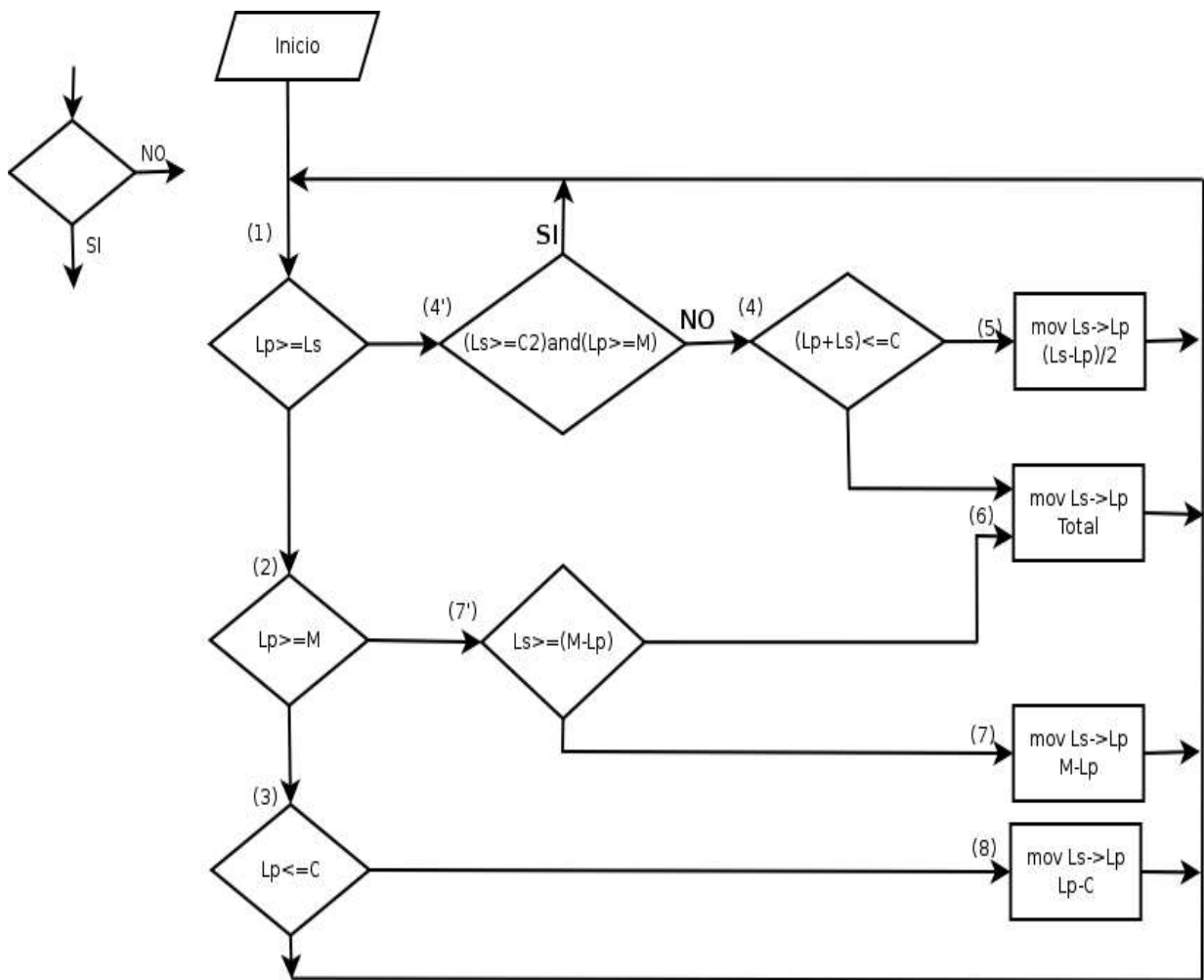
6.2.2 Algoritmo LCM Modificado

Con el objetivo de dar solución a las falencias que presenta el algoritmo LCM, el presente trabajo propone dar solución a las dos primeras falencias listadas anteriormente, realizando las siguientes modificaciones y dando como resultado el algoritmo LCM-Modificado o LCM-M.

La primera modificación en el algoritmo LCM está en (2), y consiste en que antes de ir a (7) deberá ir a (7'), donde verificará si el tráfico a mover está disponible en Ls, si NO va a (6) como se muestra en la Figura 13. Con esta modificación se evita que el algoritmo LCM entre en el bucle infinito, dando solución a la primera deficiencia detectada.

La segunda modificación en el algoritmo LCM esta en (1'), y consiste que antes de ir a (4) deberá ir a (4'), para detectar si el enlace secundario esta en saturación y si la carga en el enlace primario es mayor que M, de tal forma que si se cumplen estas dos condiciones el mecanismo detecte que los enlaces están congestionados y por lo tanto no debe balancear más carga y va a (1), si no continua con el balanceo de carga y va a (4) como se muestra en la Figura 13. Esta modificación soluciona la segunda falencia detectada en el algoritmo LCM, evitando que el tráfico oscile entre el LSP primario y el LSP secundario.

Figura 13: Diagrama de flujo del algoritmo LCM Modificado



6.2.3 Algoritmo LCM Generalizado

En este apartado se presenta una modificación al algoritmo de balanceo de carga LCM



modificado, la cual pretende adicionar un modulo que permita dar solución a la tercera falencia detectada, es decir solucionar los problemas de escalabilidad que presenta el algoritmo LCM-M, como se muestre en la Figura 14. A continuación se presenta el análisis del algoritmo de balanceo de carga LCM-M junto con el Modulo de Generalización.

(I). Crear lista de LSPs paralelos de un origen a un destino, asociando el estado actual de cada uno de ellos.

(II). Actualizar lista de los LSP en congestión. $listaLSPc = \{LSPc1, LSPc2, \dots, LSPcn\}$ y una lista con los LSPs que tengan ancho de banda disponible. $listaLSPf = \{LSPf1, LSPf2, LSPf3, \dots, LSPfn\}$. Donde $LSPcn$, $LSPfn$ son los LSPs de menor prioridad.

(III). Si $listaLSPc$ es igual a Null, ir a (VI), si no ir a (IV).

(IV). Si la $listaLSPf$ es igual a Null, ir a (I), si no ir a (V).

(V). Determinar si la capacidad disponible en $LSPf1$ es suficiente para balancear la cantidad de ancho de banda excedente del $LSPc1$, evaluando la condición " $LCPC1 - C1 \leq C2 - LSPf1$ ", Si es suficiente ir a (VII), si no ir a (VII).

(VI). Seleccionar los dos LPSs (Lp y Ls), sobre los cuales se balanceara carga, y continuar en (IX).

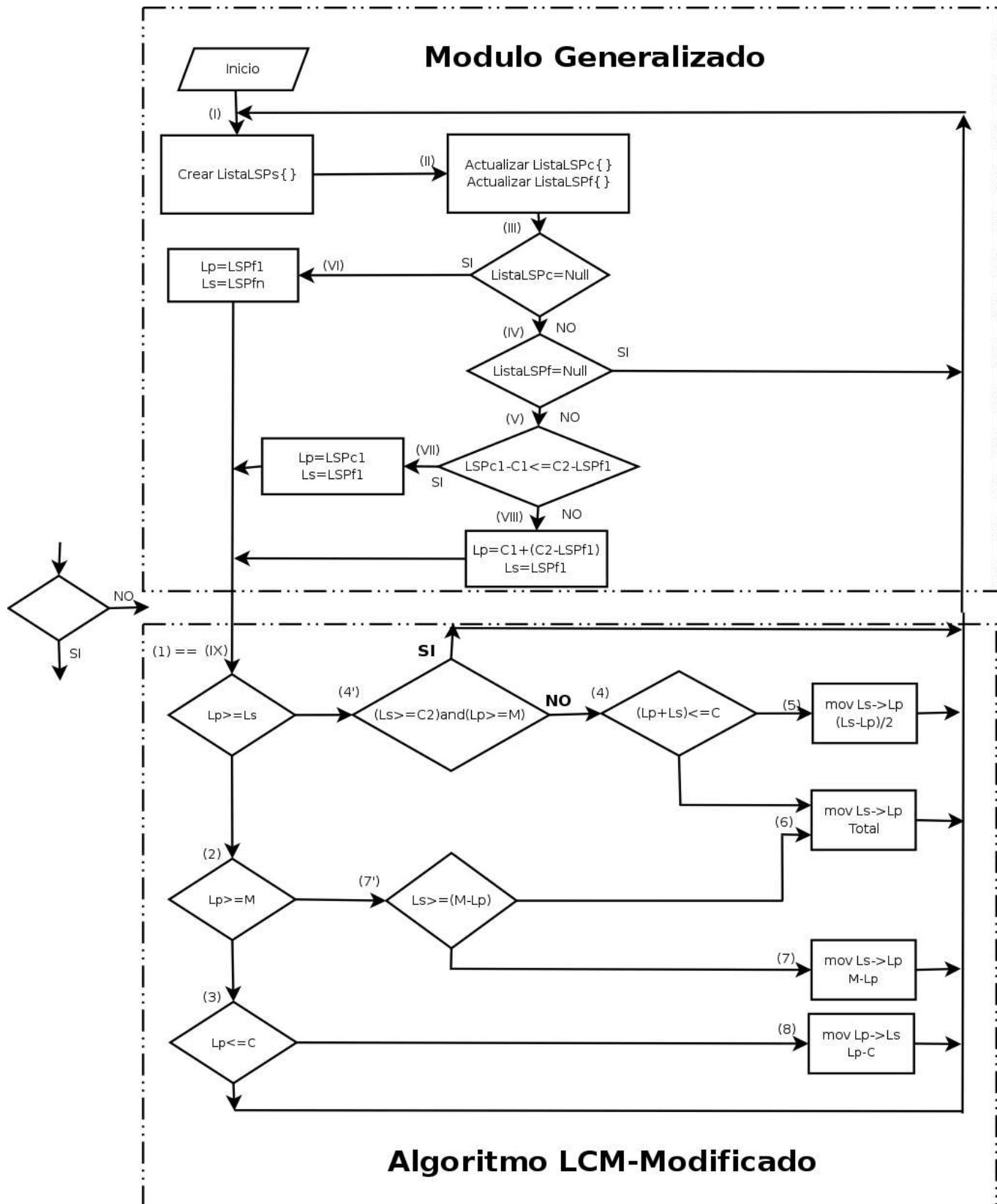
(VII). La totalidad del ancho de banda excedente de Lp puede ser balanceado sobre Ls sin llegar a saturarlo. Ir a (IX).

(VIII). Adaptar la cantidad de ancho de banda a balancear, intentando evitar que el LSP secundario pueda ser saturado. Luego va a (IX) $==(1)$.

(IX). Llamar el algoritmo LCM-Modificado.



Figura 14: Algoritmo LCM Generalizado





6.3 VALIDACION DEL MECANISMO DLCCM

En este apartado se dará una explicación general de la pre-configuración de la red MPLS experimental. Los archivos de configuración de quagga, rsvpd, configuración del kernel y la guía paso a paso de la implementación de la red MPLS experimental se encuentran en el anexo B. También se diseñarán y ejecutarán las pruebas necesarias para validar el funcionamiento del mecanismo de balanceo de carga implementado.

6.3.1 Pre-Configuración de la Red MPLS Experimental

Una de las limitantes en la validación del mecanismo de balanceo de carga, es la disponibilidad de recursos hardware, por esta razón se utilizó el software VMWare [40], el cual permite mediante la técnica de virtualización obtener múltiples sistemas operativos en una misma máquina; por consiguiente la primera tarea que se debe ejecutar consiste en instalar los sistemas operativos en la máquina virtual VMWare y configurar diferentes segmentos de red que permitan implementar el diagrama de red mostrado en la figura 10. De esta manera se tendrán disponibles tanto los LERs como los LSRs.

Los LSRs tienen el sistema operativo redhat 9.0, sobre él se instaló quagga-0.99.4 como software de enrutamiento, habilitándole las extensiones para Ingeniería de tráfico al protocolo OSPF-TE, también tienen instalado rsvpd.0.70-rc2, el cual implementa el protocolo RSVP-TE al igual que el estándar MPLS. La razón por la cual se utilizó redhat 9.0 para la implementación del backbone MPLS siendo este un sistema operativo un poco discontinuado, es que el demonio rsvpd funciona adecuadamente en redhat, pruebas realizadas en fedora core 4, debian 3.0 y 3.1 no produjeron resultados satisfactorios.

Los demonios se habilitan de la siguiente manera:

Primero se debe habilitar ospfd y luego zebra, finalmente se habilita rsvpd.

```
[root@sarge root]# /etc/init.d/ospfd start
```

```
[root@sarge root]# /etc/init.d/quagga start
```

```
[root@sarge root]# rsvpd -D
```

Adicionalmente se debe correr el proceso que implementa las funciones de colección y



distribución, las cuales están implementadas en el archivo “colectorV05” y se ejecuta de la siguiente manera:

```
[root@sarge root]# /root/bin/./colectorV05 &
```

Los LERs corren el mismo software que los LSRs, pero adicionalmente los LERs deben tener preconfigurados los LSPs explícitos, lo cual se hace utilizando una de las extensiones de ingeniería de tráfico disponible en RSVP-TE e implementada en el demonio rsvpd. Haciendo uso de las utilidades disponibles en el demonio rsvpd, se debe habilitar el LER de egreso para que una vez reciba un mensaje “PATH” envíe automáticamente el mensaje “RESV”, como se muestra a continuación:

En el LER de egreso:

```
[root@puffy root]# rapirecv_auto &
```

En el LER de Ingreso se deben enviar los mensajes “PATH” solicitando la creación de los LSPs, mediante la utilidad “rtest2”.

```
[root@puffy root]# rtest2 -f /root/bin/file
```

El archivo “file” define los LSPs explícitos, así pueden ser creados varios LSPs simultáneamente, con un simple comando. El formato de configuración de los LSP en el archivo “file” es el siguiente:

```
<dst_ip> <src_ip> <lsp_id> <PHBID> <IntServ> <hop_ip1:hop_id2:hop_ip3:0> <SE style>
```

dst_ip: Dirección IP de destino.

src_ip: Dirección IP de Origen.

lsp_id: Identificador del LSP.

PHBID: Comportamiento salto a salto, utilizado generalmente con Diff -Serv.

IntServ: Define el uso de señalización en servicios integrados.

hop_ip[1,2,3]: Dirección IP de cada salto que conforma el LSP explícito, el máximo número de saltos soportado es 19.



SE style: Define el tipo de reserva Explícitamente Compartido (SE:Shared Explicit) para el LSP, el cual le permite hacer una reserva simple, pero el receptor puede identificar qué transmisores podrían compartir dicha reserva.

El contenido del archivo “file” es el siguiente:

```
192.168.206.1 192.168.201.1 200 0 0 192.168.201.2:192.168.203.1:0 0  
192.168.206.1 192.168.202.1 201 0 0 192.168.202.2:192.168.204.1:192.168.205.1:0 0
```

Luego de creados los LSPs se debe ejecutar el archivo “lbmplsV10”, con lo cual finaliza la implementación y puesta en funcionamiento del mecanismo de balanceo de carga en redes MPLS.

```
[root@puffy bin]# ./lbmplsV10 &
```

6.3.2 Tipo de Tráfico Utilizado

El tipo de tráfico que se utilizará será tipo Poisson, puesto que es un modelo utilizado tradicionalmente para describir el tráfico en Internet [41]. El tráfico Poisson será suficiente para enviar tráfico a la red MPLS experimental y así validar el funcionamiento del mecanismo DLCCM, para este propósito se utilizará una herramienta desarrollada bajo software libre llamada mgen [42], que permite generar múltiples flujos de datos de tipo constante y tipo Poisson.

El formato utilizado por la herramienta mgen para la creación de los flujos es el siguiente:

```
Tiempo ejecución—Flujo ID—Conmutador--IP destino:puerto--Tipo Tráfico—Nº  
paquetes—Tamaño de Paquete  
00000 1 ON 192.168.206.2:5000 POISSON 12 200
```

Tiempo de ejecución: Tiempo en el cual se debe ejecutar la sentencia.

Flujo ID: ID (nombre) del flujo, que por lo general es un número.

Conmutador: Indica si el flujo se va a adicionar o si se va a detener, con la sentencia ON o OFF respectivamente.



IP Destino: puerto: Indica la IP y el puerto de destino a donde se va a enviar el flujo de tráfico.

Tipo de Tráfico: Tipo de Tráfico que se va a cursar, el cual puede ser: Constante y Poisson.

Nº de Paquetes: indica el número de paquetes que se van a enviar en este flujo de datos.

Tamaño de Paquete: Tamaño de cada paquete que se va a enviar.

6.3.3 Diseño de las Pruebas

Acorde a la red implementada mostrada en la Figura 10, se realizarán tres tipos de pruebas, en dos escenarios diferentes en los cuales se pretende mostrar el funcionamiento y las reacciones del mecanismo de balanceo de carga ante variaciones en el tráfico incidente en el dominio MPLS experimental.

El primer escenario evalúa el funcionamiento del mecanismo de balance de carga insertando tráfico desde una sola fuente. La primera fuente ingresa tráfico por medio del Host A (ver figura 10), ingresando tráfico a través del LER 1, teniendo como destino el Host B. De ahora en adelante cuando se hable de fuente 1 se hará referencia al tráfico que genera el Host A.

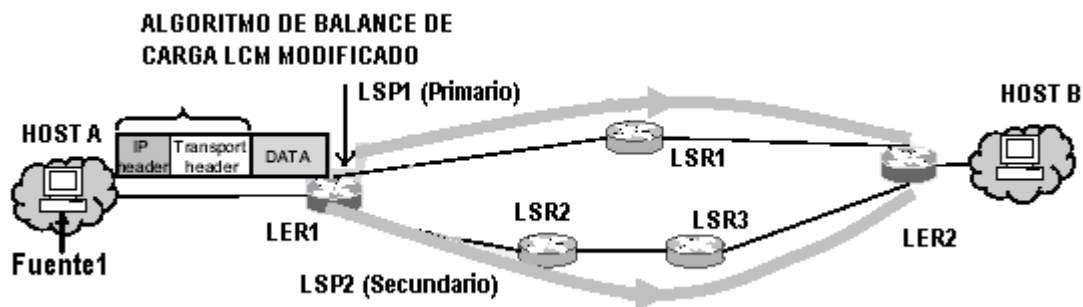
En el segundo escenario se pretende simular tráfico intermedio (Cross Traffic), tráfico que se inserta en el core MPLS, con el fin de observar cómo reacciona el mecanismo de balance de carga cuando existe tráfico cruzado, y ver la influencia de este tráfico en el estado actual de la red. De ahora en adelante cuando se hable de fuente2 se hará referencia al tráfico que se genera en el core MPLS, más exactamente desde el LSR1 (ver figura 10)

En los dos escenarios se realizarán medidas en periodos de 10 segundos, datos que serán utilizados para graficar el tráfico como una variable del tiempo. Las medidas serán realizadas en el LER1 sobre los dos LSPs configurados y sobre el LSR1 para determinar la cantidad de tráfico cruzado que ingresa al dominio MPLS y cual es el impacto que genera en el mecanismo de balanceo de carga implementado en el LER1.

6.3.3.1 Escenario 1

En el escenario uno (ver figura 15) se ingresará tráfico a la red MPLS a través de una sola fuente de tráfico, que es la fuente1.

Figura 15: Escenario 1, Una sola Fuente de Tráfico



6.3.3.1.1 Prueba 1A

En la prueba 1A se muestra el funcionamiento básico del mecanismo de balanceo de carga, con el fin de tomarlo como referencia en un análisis posterior.

Las características de esta prueba son:

- Umbral C= 70% y Umbral M= 50%.
- Un solo tipo de tráfico.

6.3.3.1.1.1 Cronograma de la Prueba 1A

El siguiente es el cronograma de pruebas utilizado, el cual tiene una sola fuente de tráfico. Únicamente se inyecta tráfico desde el Host A, el cual atraviesa el LER1 y tiene como destino el Host B.

Es de aclarar que los flujos de tráfico insertados no varían a través del tiempo.

- Minuto 0. Inyectar tráfico a la red MPLS experimental, sin que este sobrepase los



límites de congestión (C).

- Minuto 5. Incrementar el tráfico hasta superar los umbrales definidos como congestión (C).
- Minuto 10. Incrementar nuevamente el tráfico hasta superar los umbrales definidos como congestión (C).
- Minuto 15. Reducir el tráfico hasta hacerlo igual o menor al umbral de trabajo (M).
- Minuto 25 Terminar el tráfico total.

De acuerdo a la sentencia ejecutada en *mgen* mostrada a continuación se puede generar tráfico por flujos:

```
00000 1 ON 192.168.206.2:5000 POISSON 12 200
```

12 paquetes * 245 bytes/paquete * 8bits/byte * 1 flujo = +/-23kbps. Por ser tráfico Poisson.

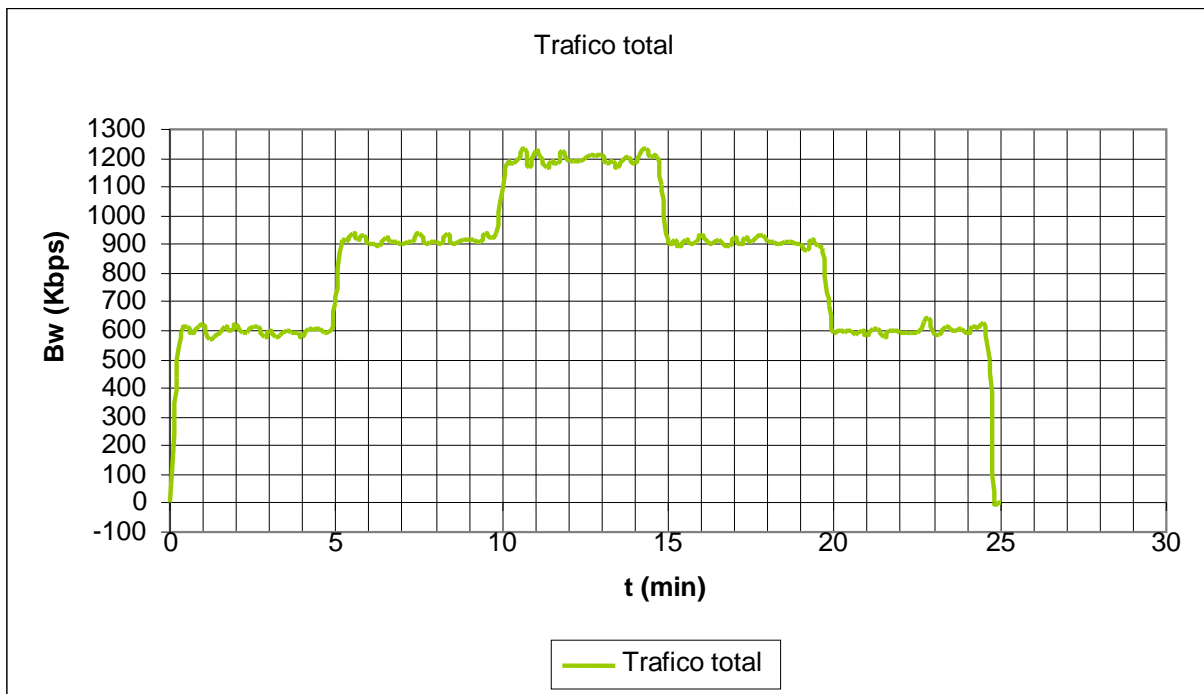
Nota: El tamaño de cada paquete mostrado en la anterior sentencia de *mgen* es de 200 bytes, pero en realidad el tráfico que se envía es 245 bytes ya que antes no se tenía en cuenta los encabezados de los protocolos de nivel inferior.

Tabla 3: Tráfico Prueba 1A

<i>Tiempo(min)</i>	<i>Nº Flujos</i>	<i>BW Total</i>	<i>Tráfico LER1</i>	<i>Tráfico LSR1</i>
0	25	588 Kbps	25	0
5	38	893 Kbps	13	0
10	50	1176 Kbps	12	0
15	38	893 Kbps	-12	0
20	25	588 Kbps	-13	0
25	0	0 Kbps	-25	0

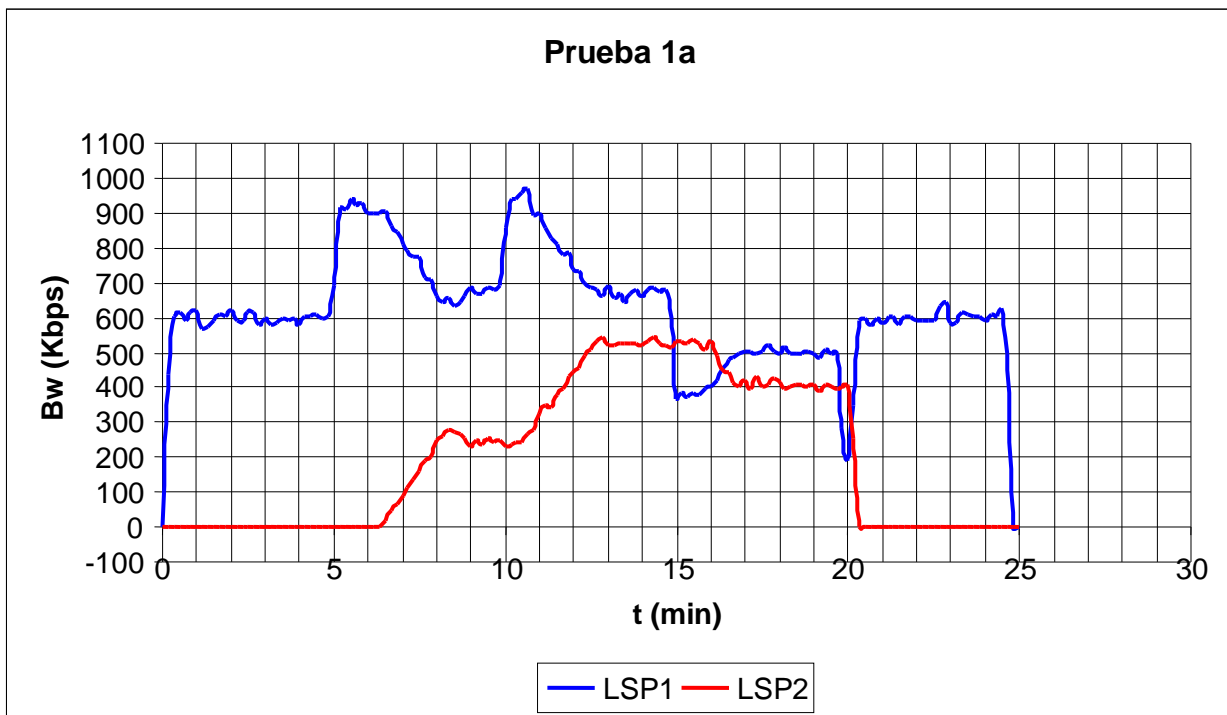


Figura 16: Tráfico Total Prueba 1A



6.3.3.1.1.2 Resultados de la Prueba 1A

Figura 17: Resultados Prueba 1A





6.3.3.1.1.3 Análisis de Resultados, Prueba 1A

- **Minuto 5:** En el minuto 5 hay un incremento de tráfico en el LSP1, que supera el umbral “C”, con lo cual el mecanismo de balanceo de carga propuesto detecta estado de congestión, produciendo balanceo de carga sobre el enlace secundario. La cantidad de ancho de banda balanceada depende del estado actual de los LSPs, en este caso se balanceará una cantidad $(L_p - C)$, $L_p = 950\text{Kbps}$ y $C = 700\text{Kbps}$. Por lo tanto la cantidad de tráfico a balancear sería de 250Kbits aproximadamente. En la figura 17 se puede observar que los 250Kbits son balanceados hacia el LSP2.

De acuerdo al diagrama de flujos mostrado en la Figura 13, la secuencia de estados es la siguiente:

Inicia en (1) y evalúa la sentencia $L_p \geq L_s$, si es verdadera va (2) donde evalúa si $L_p \geq M$, como es verdadero va a (3), puesto que $L_p > C$ entonces va a ejecutar el proceso en (8), que consiste en balancear carga de $L_p \rightarrow L_s$ una cantidad $L_p - C$.

Analizando la Figura 17 se puede observar que el tiempo utilizado para balancear el tráfico $(L_p - C)$ es de 3 min aproximadamente, el cual depende de la cantidad de tráfico por flujo, como se mencionó en 6.1.2.5 entre menor sea la cantidad de tráfico por flujo, mayor será el tiempo utilizado para balancear determinada cantidad de tráfico, pero será mayor la precisión del balanceo.

Otra cosa que se puede analizar de la figura 17 es que el LSP1 se estabiliza en un valor cercano a 700Kbps, ya que este es el valor del umbral de congestión C.

- **Minuto 10:** Sucede algo similar a lo ocurrido en el minuto 5, solo que el tiempo que se necesita para balancear es de 2.8 min aproximadamente.
- **Minuto 15:** En el minuto 15 se puede observar que hay un decremento en el flujo de tráfico de 300Kbps aproximadamente, esto hace que la carga del enlace primario sea menor que la carga del enlace secundario ($L_p = 400\text{Kbps} < L_s = 500\text{Kbps}$), produciendo un balanceo de carga de $(L_s - L_p)/2$ del LSP2 hacia el LSP1 hasta que L_p sea mayor a L_s , cuando esto suceda el algoritmo sigue corriendo y detecta que L_p es menor que M , produciendo de nuevo un balanceo de carga de $M - L_p$, de tal manera que el tráfico



en el LSP1 se estabiliza en M, cuyo valor es 500Kbps.

De acuerdo al diagrama de flujos mostrado en la Figura 13, la secuencia de estados es la siguiente:

Va a (1) evalúa la sentencia $L_p \geq L_s$ y retorna falso. Va a (4') evalúa la sentencia $(L_s \geq C_2) \text{ and } (L_p \geq M)$ y retorna falso. Va a (4) evalúa la sentencia $(L_p + L_s) < C$ y retorna falso. Va a (5) y ejecuta balancear carga de $L_s \rightarrow L_p$ una cantidad $(L_s - L_p)/2$. Después de esto el algoritmo sigue corriendo y va a (1) evalúa $L_p \geq L_s$ y retorna verdadero. Va a (2) evalúa la sentencia $L_p \geq M$ y retorna falso. Va a (7') y evalúa la sentencia $L_s \geq (M - L_p)$ y retorna verdadero. Va a (7) y ejecuta balancear carga de $L_s \rightarrow L_p$ una cantidad $M - L_p$.

- **Minuto 20:** En el minuto 20 podemos, observar que hay un decremento en el flujo de tráfico de 300Kbps aproximadamente, esto hace que la carga del enlace primario sea menor que la carga del enlace secundario ($L_p = 200\text{Kbps} < L_s = 400\text{Kbps}$), produciendo un balanceo de carga total del enlace secundario hacia el primario debido a que todo el tráfico actual puede ser cursado por el LSP1. En la Figura 17 se puede observar que la carga del LSP2 (400Kps) es balanceada totalmente hacia el enlace Primario, llegando este último a transportar los 600Kbps que la fuente1 esta ingresando a la red.
- **Minuto 25:** En este minuto el tráfico total es terminado, observándose en la figura 17 que los enlaces están vacíos.

6.3.3.1.2 Prueba 2A

En esta prueba se muestra el funcionamiento del algoritmo de balanceo de carga cuando el tamaño de los flujos de tráfico varía.

Las características de esta prueba son:

- Umbral C= 70% y Umbral M= 50%
- Se trabajará con 3 tipos de tráfico de diferente tamaño.



6.3.3.1.2.1 Cronograma de la Prueba 2A

En la siguiente prueba se enviará tráfico desde una sola fuente, donde la cantidad de tráfico por flujos varía. Únicamente se inyecta tráfico desde un host externo PC1, el cual atraviesa el LER1 y tiene como destino el host "PC2".

Se enviaran 3 tipos de tráfico, cada uno con cantidad de tráfico diferente, donde $\text{tipo1} < \text{tipo2} < \text{tipo3}$

- Minuto 0. Inyectar tráfico a la red MPLS experimental, sin que este sobrepase los límites de congestión (C), inyectando tráfico de tipo 1.
- Minuto 5. Incrementar el tráfico hasta igualar o superar los umbrales definidos como congestión (C). inyectando tráfico de tipo 2.
- Minuto 10. Incrementar nuevamente el tráfico hasta superar los umbrales definidos como congestión (C). inyectando tráfico de tipo 3.
- Minuto 15. terminar el tráfico de tipo 3 sin que la cantidad de tráfico total sea menor al umbral de congestión (C).
- Minuto 20. Terminar el tráfico de tipo 2 haciendo que la cantidad de tráfico total sea menor al umbral de congestión (C).
- Minuto 25 Terminar el tráfico total.

De acuerdo a las sentencias ejecutadas en *mgen*, descritas anteriormente, se pueden generar distintos tipos de tráfico con diferente tamaño:

tipo 1 = +/-60.7kbps.

tipo 2 = +/-100kbps

tipo 3 = +/-150kbps



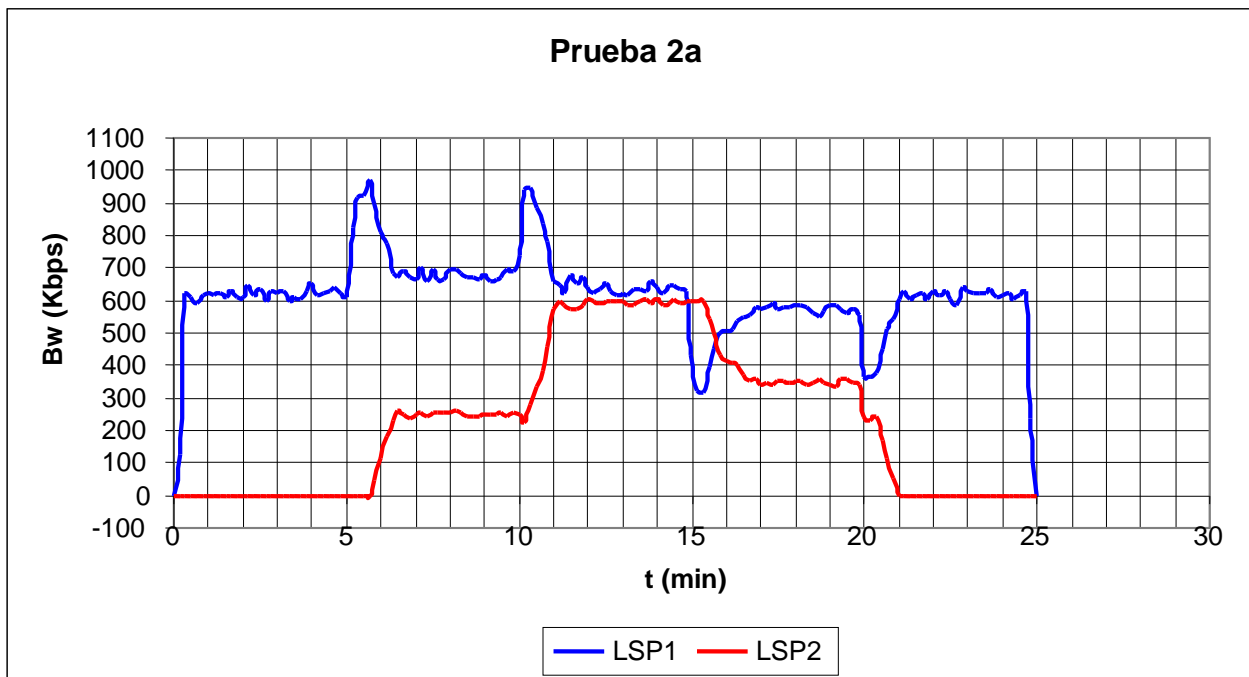
Tabla 4: Tráfico Prueba 2A

<i>Tiempo(mins)</i>	<i>N° Flujos</i>	<i>BW Total</i>	<i>Tráfico LERI</i>	<i>Tráfico LSRI</i>
0	10	600	10	0
5	13	900	3	0
10	15	1200	2	0
15	13	900	-2	0
20	10	600	-3	0
25	0	0	0	0

El tráfico total es similar al que se genero en la prueba 1A (ver figura 16)

6.3.3.1.2.2 Resultados de la Prueba 2A

Figura 18: Resultados Prueba 2A





6.3.3.1.2.3 Análisis de Resultados, Prueba 2A

- **Minuto 5:** El análisis que se puede realizar es el mismo que se hizo en el análisis de resultados de la prueba 1A, por tal razón se hará mención únicamente a las variaciones que sean relevantes en la prueba 2A.

Se observa en la Figura 18 que en el minuto 5 existe un incremento de ancho de banda que hace que el mecanismo de balanceo de carga detecte congestión, produciéndose un balanceo de carga del LSP1 al LSP2, en una cantidad similar a la que se pudo analizar en 6.3.3.1.1.3.

Es importante anotar que el tiempo que necesita el mecanismo de balanceo de carga para mover la misma cantidad de tráfico (Lp-C) del LSP2 al LSP1, es de 1.3 min aproximadamente, con lo cual se puede encontrar una diferencia con la Prueba 1A, donde el tiempo empleado era de 3min aproximadamente. Lo anterior tiene su explicación ya que de acuerdo a lo mencionado en 6.3.2.6 el mecanismo necesita menos tiempo para balancear la carga ya que los flujos de tráfico son más grandes, como ocurre en este caso.

- **Minuto 10:** En el minuto 10 ocurre algo similar que en el minuto 5, en este caso el tiempo necesitado por el mecanismo para balancear la carga es de 1min aproximadamente.

Algo que se puede apreciar en la Figura 18 es que la carga del LSP1 se estabilizó en un valor más distante del umbral de congestión C, que era al cual debería llegar en el tiempo de su estabilización. Esta imprecisión es debida al incremento en el tamaño del tráfico por flujo, ya que en algunas ocasiones al intentar balancear una pequeña porción de tráfico, el mecanismo balancea una cantidad de tráfico mayor debido a que el tamaño de sus flujos es muy grande.

- **Minuto 15:** En el minuto 15 se remueve una cantidad de tráfico entrante aproximadamente de 300Kbps, lo cual causa el mismo efecto ya analizado en la Prueba 1A, en este caso se puede notar una impresión en el balanceo de carga desde el LSP2 hacia el LSP1, puesto que el valor de estabilización debió ser el umbral



“M” y no estar tan alejado de este, como ocurre en la Figura 18, esto se debe a las imprecisiones que se presentan en los mecanismos de balanceo de carga mediante división por flujos, cuando el ancho de banda por cada flujo es grande.

- **Minuto 20:** El efecto observado en el minuto 20 corresponde al mismo ya analizando en el mismo intervalo de tiempo de la Prueba 1A.
- **Minuto 25:** En este minuto es terminado el tráfico total, observándose en la Figura 18 que los enlaces están vacíos.

6.3.3.1.3 Prueba 3A

La Prueba 3A consiste en variar los umbrales de congestión de tal manera que sean más altos, esperando así que se obtenga una mayor utilización de los enlaces.

En esta prueba se varía el tráfico total enviado a la red MPLS para que el nuevo tráfico no sobresature el enlace, ya que la máxima capacidad por LSP es de 1M y el umbral de congestión “C” es de 90% del máximo, lo que hace que la diferencia entre la máxima capacidad del enlace y “C” sea de 100Kbps, por tanto cuando se presente balanceo de carga el enlace principal queda cercano al umbral de congestión, haciendo que una cantidad de tráfico superior a 100Kbps sobresature el enlace.

Lo que busca esta prueba es mostrar que el administrador puede variar los umbrales de congestión, dependiendo de la máxima utilización que necesite en los enlaces.

Las características de esta prueba son:

- Umbral C= 90% y Umbral M= 70%
- Se trabajará con un solo tipo de tráfico.

6.3.3.1.3.1 Cronograma de la Prueba 3A

Se variarán los umbrales C y M, que antes estaban en 70% y 50% incrementándolos a 90% y 70% respectivamente.



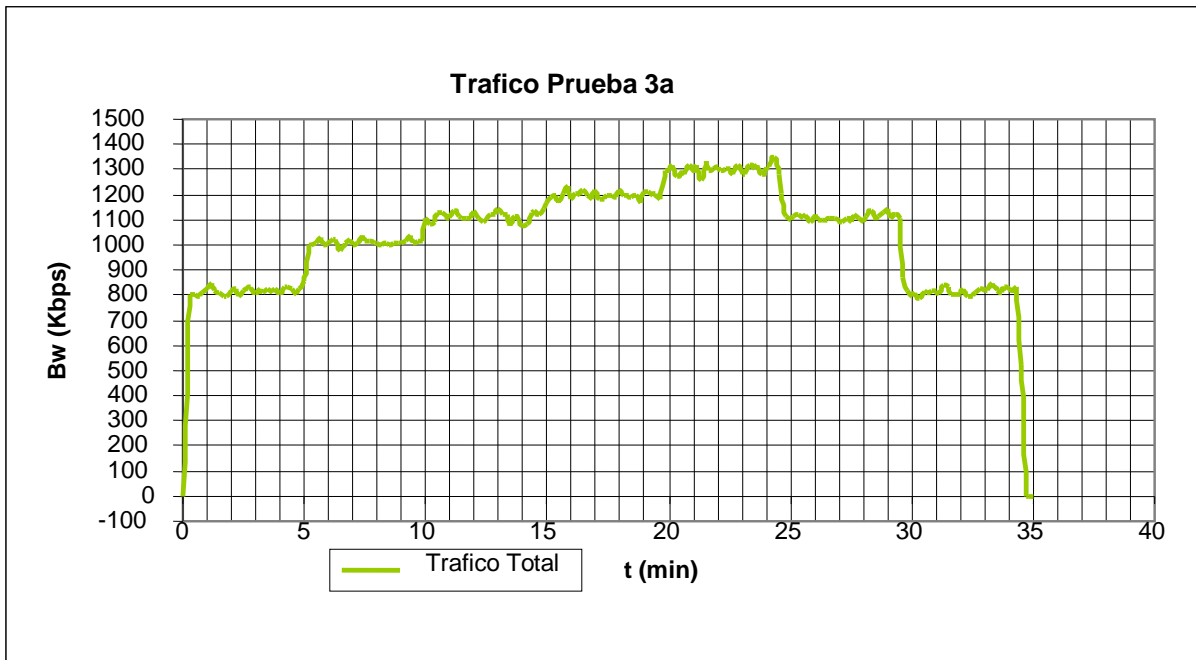
- Minuto 0. Inyectar tráfico a la red MPLS experimental, sin que este sobrepase los límites de congestión (C).
- Minuto 5. Incrementar el tráfico hasta igualar o superar los umbrales definidos como congestión (C).
- Minuto 10 15 y 20. Incrementar nuevamente el tráfico hasta superar los umbrales definidos como congestión (C).
- Minuto 25. Terminar el tráfico que se inicio en el minuto 15 y 20, sin que la cantidad de tráfico total sea menor al umbral de congestión (C).
- Minuto 30. Terminar el tráfico que se inicio en el minuto 5 y 10.
- Minuto 35. Terminar el tráfico total.

Tabla 5: Tráfico Prueba 3A

<i>Tiempo(mins)</i>	<i>N° Flujos</i>	<i>BW Total</i>	<i>Tráfico LER1</i>	<i>Tráfico LSR1</i>
0	34	800	34	0
5	42	988	8	0
10	46	1082	4	0
15	50	1176	4	0
20	54	1270	4	0
25	46	1082	-8	0
30	34	800	-12	0
35	0	0	-34	0

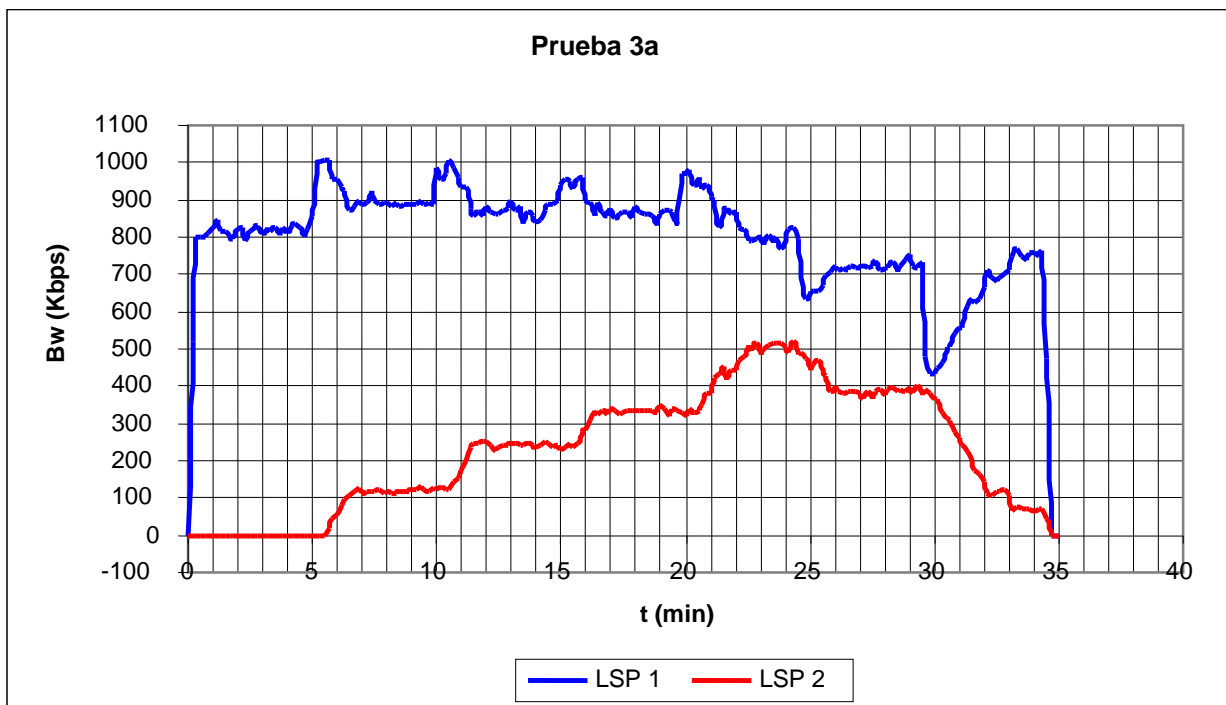


Figura 19: Tráfico Total Prueba 3A



6.3.3.1.3.2 Resultados de la Prueba 3A

Figura 20: Resultados Prueba 3A





6.3.3.1.3.3 Análisis de Resultados, Prueba 3A

El análisis que se puede realizar es el mismo que se hizo en la prueba 1A, la diferencia radica básicamente en que en el minuto 0 se enviaron 800Kbps, que tendrían el mismo efecto de los 600Kbps enviados en la prueba 1A, dado que ahora el umbral de congestión “C” se fija en 900Kbps, con el cual no se produce ningún tipo de balanceo.

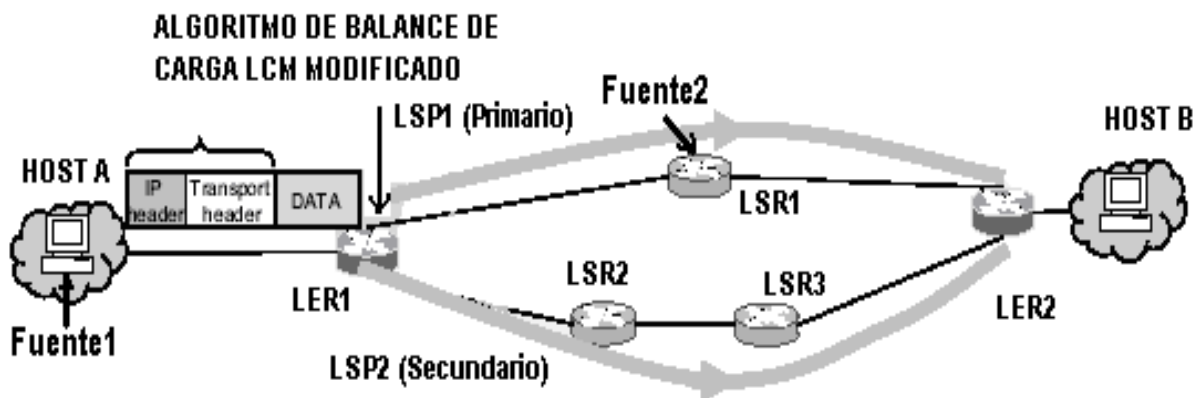
- **Minutos 5,10,15, y 20:** se envían pequeñas cantidades de tráfico aproximadamente de 100Kbps en cada una de ellas, con las cuales el sistema entra en congestión y balancea una carga de tráfico del primario al secundario aproximadamente de L_p-C , en cada uno de los tiempos mencionados anteriormente. Se puede notar que en el minuto 20 la cantidad de tráfico que el algoritmo LCM balancea, es mayor a L_p-C , puesto que el valor de estabilización se sitúa alrededor de 800Kbps y no de 900Kbps como sucede en los minutos 5,10, y 15. Este comportamiento anormal se debe a que el ambiente de implementación de la red MPLS experimental, como ya se ha mencionado en apartados anteriores, esta implementado bajo un sistema de máquinas virtuales alojadas en una única máquina física, y puesto que el tráfico enviado a la red MPLS experimental es real, este causa que el rendimiento de la máquina en la cual se tienen configuradas las máquinas virtuales disminuya notablemente y las medidas realizadas para calcular el estado actual de los enlaces presenten imprecisiones considerables.
- **Minuto 25:** En este minuto se termina una cantidad de tráfico aproximadamente de 200Kbps, lo cual hace que el algoritmo LCM realice balanceo de carga desde el secundario con destino al primario hasta que el tráfico en el primario se estabilice alrededor del umbral “M”.
- **Minuto 30:** En este minuto se termina una cantidad de tráfico aproximadamente de 300Kbps, lo cual hace que el algoritmo LCM realice balanceo de la carga total que tiene el LSP2 hacia el LSP1, puesto que todo el tráfico puede ser transportado por el enlace principal (LSP1).
- **Minuto 35:** Es finalizado la totalidad del tráfico entrante por lo cual los enlaces LSP1 y LSP2 quedan vacíos y el algoritmo LCM queda en reposo.

6.3.3.2 Escenario 2

En el escenario 2 (ver figura 21) se ingresa tráfico a través de dos fuentes, fuente 1 y fuente2, con el fin de simular el Cross Traffic o Tráfico cruzado que ingresa a la red por los nodos intermedios.

Los resultados que se esperan obtener cuando se ingresa Cross Traffic han de ser similares a los que se obtuvieron en el escenario 1, por tal motivo no se hará un análisis exhaustivo de lo que pasa en cada cambio de tráfico, simplemente se espera comprobar que el algoritmo funciona correctamente aun cuando el tráfico actual de la red MPLS depende del Cross Traffic que hay en el core de la red.

Figura 21: Escenario 2, Dos Fuentes de Tráfico



6.3.3.2.1 Prueba 1B

En la Prueba 1B se pretende mostrar el funcionamiento básico del mecanismo de balanceo de carga, cuando se inyecta cross traffic, con el fin de tomarlo como referencia en un análisis posterior.

Las características de esta prueba son:

- Umbral C= 70% y Umbral M= 50%
- Un solo tipo de tráfico.



6.3.3.2.1.1 Cronograma de la Prueba 1B

Es de aclarar que los flujos de tráfico insertados no varían a través del tiempo.

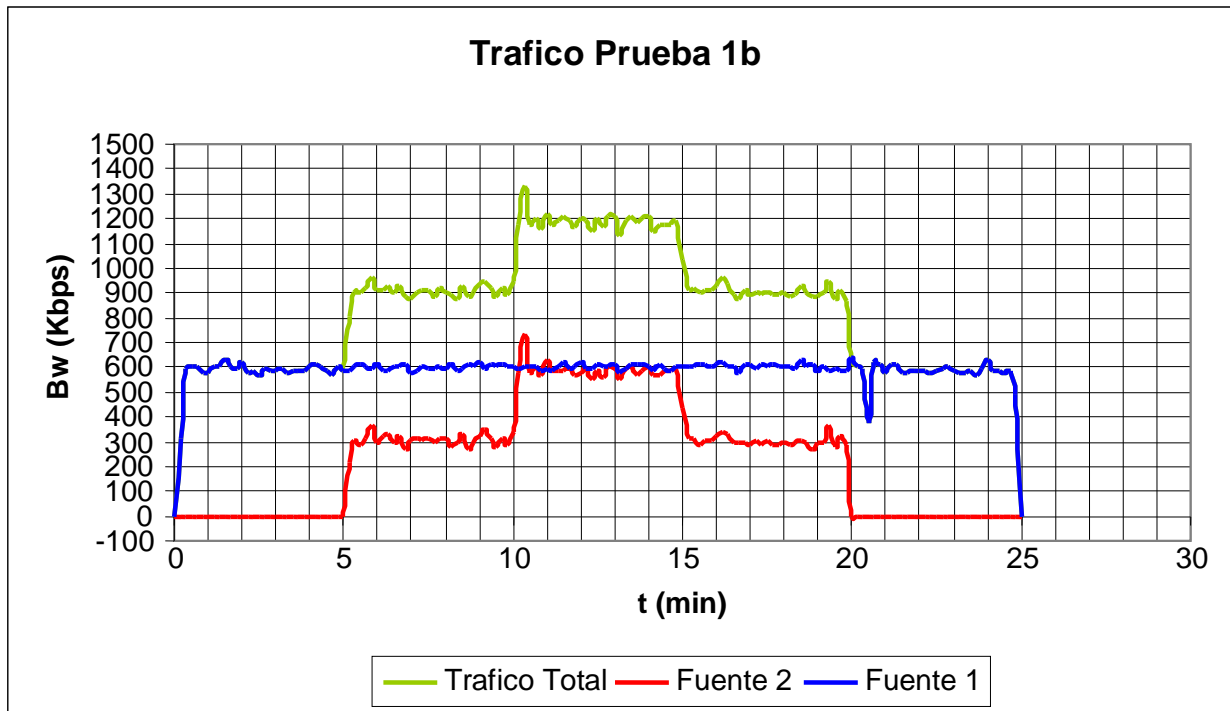
- Minuto 0. Inyectar tráfico desde la fuente 1 (ingresa por el host PC1 y atraviesa el LER1), a la red MPLS experimental, sin que este sobrepase los límites de congestión (C), por un periodo de 5 minutos.
- Minuto 5. Incrementar el tráfico desde la fuente 2 (LSR1) hasta igualar o superar los umbrales definidos como congestión (C) y mantenerlo por un periodo de 5 minutos.
- Minuto 10. Inyectar nuevamente tráfico desde la fuente 2 (LSR1), de tal manera que se supere el umbral de congestión (C) y mantenerlo por un periodo de 5 minutos.
- Minuto 15. Terminar la última cantidad de tráfico generado por la fuente 2 (LSR1).
- Minuto 20. Terminar la segunda cantidad de tráfico generada por la fuente 2 (LSR1).
- Minuto 25. Terminar el tráfico generado por la fuente 1.

Tabla 6: Tráfico Prueba 1B

<i>Tiempo(mins)</i>	<i>Nº Flujos</i>	<i>BW Total</i>	<i>Tráfico LER1</i>	<i>Tráfico LSR1</i>
0	25	588 Kbps	25	0
5	38	893 Kbps	0	13
10	50	1176 Kbps	0	12
15	38	893 Kbps	0	-12
20	25	588 Kbps	0	-13
25	0	0 Kbps	-25	0

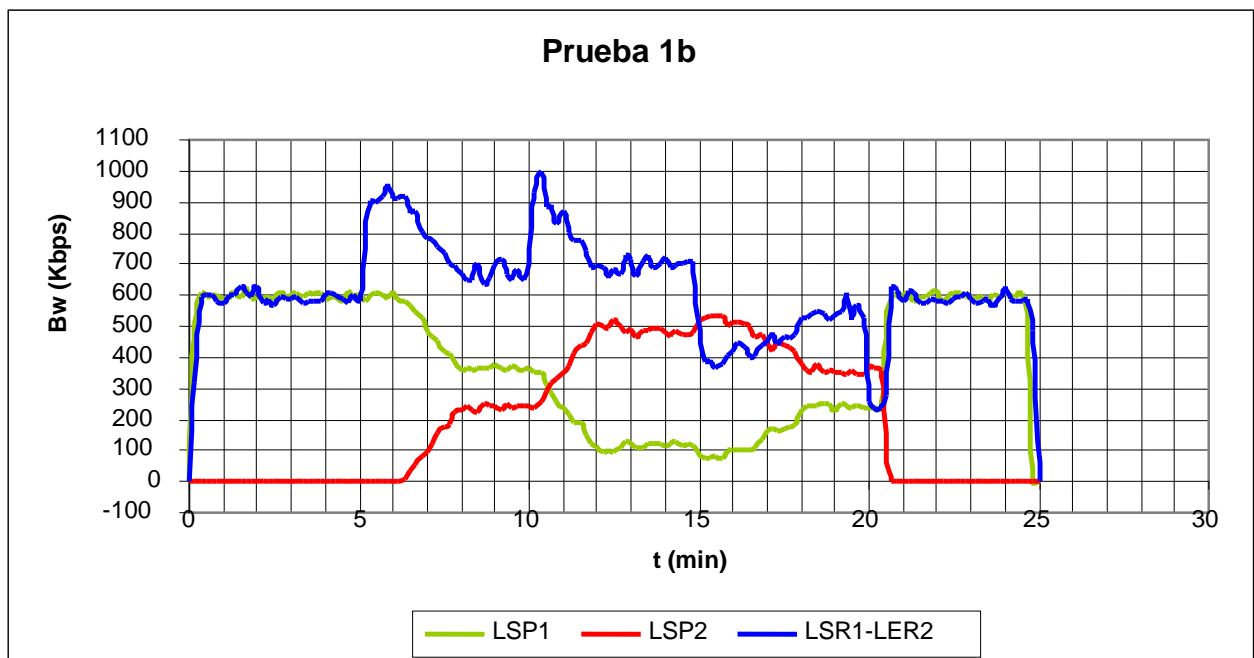


Figura 22: Tráfico Total Prueba 1B



6.3.3.2.1.2 Resultados de la Prueba 1B

Figura 23: Resultados Prueba 1B





6.3.3.2.1.3 Análisis de Resultados, Prueba 1B

- **Minuto 0:** En la Figura 23 se observa que la fuente 1 inicia la generación de tráfico destinado al hostB, aproximadamente de 600Kbps. Puesto que el umbral de congestión "C" se fijo en 700Kbps, no se producirá ningún tipo de balanceo de carga.
- **Minuto 5:** La fuente 2 inicia la generación de tráfico destinado al host B aproximadamente de 300Kbps, el cual causa congestión en el tramo comprendido entre el LSR1 y LER2, denominado en adelante como "LSR1-LER2". EL tramo LSR1-LER2 será el cuello de botella que tendrá el LSP1 y que mediante la función colectora y las extensiones TE de OSPF será detectado y transmitido hasta el nodo de Ingreso o LER1. El algoritmo LCM modificado detecta el cuello de botella aproximadamente 1min después de que este ha ocurrido, esto se debe principalmente al retardo de desplazamiento que sufren los LSAs Opacos generados en el Core, para llegar al borde de la red MPLS, sumado a otros retardos implícitos del mecanismo de carga implementado, este retardo puede ser variable debido a la característica asincrónica de los procesos colectores implementadas en el core de la red y el algoritmo LCM modificado implementado en el LER1. Una vez detectado el estado de congestión por el LER1, ejecuta el algoritmo LCM modificado y balancea una carga aproximada de $L_p-C=250Kbps$ sobre el LSP2 como se muestra en la Figura 23.

El tiempo necesario para balancear la carga es de 3 min aproximadamente, el cual es un tiempo similar al que se empleo en la prueba 1A cuando no se tenia Cross traffic sobre la red.

- **Minuto 10:** En el minuto 10 ocurre algo similar que en el minuto 5, en este caso el tiempo necesitado por el mecanismo para detectar el estado de congestión es despreciable.
- **Minuto 15:** En el minuto 15 se remueve una cantidad de tráfico aproximada de 300Kbps de la fuente 2, aumentando la cantidad de ancho de banda disponible en el LSP1, la cual es detectada en el LSR1 y difundida a la red por los LSA-opacos, y detectado por LER1 un minuto después por los efectos mencionados anteriormente, cuando esto ocurre el algoritmo LCM modificado balancea carga del LSP2 al LSP1



hasta estabilizarse alrededor del umbral M de la misma forma como se ha explicado en Pruebas anteriores.

- **Minuto 20:** En este minuto se elimina totalmente el cross traffic produciéndose un balanceo total del LSP2 al LSP1, retornando al estado inicial cuando solo se ingresaba tráfico por la fuente1
- **Minuto 25:** En este minuto se termina el tráfico total, observándose en la Figura 23 que los enlaces están vacíos.

6.3.3.2.2 Prueba 2B

Esta prueba es similar a la prueba 3, se envía tráfico de diferente tamaño, pero esta vez se hace desde dos fuentes diferentes, ingresando Cross Traffic.

- Umbral C= 70% y Umbral M= 50%
- Se trabajará con 3 tipos de tráfico de diferente tamaño.

6.3.3.2.2.1 Cronograma de la Prueba 2B

Se enviarán 3 tipos de tráfico, cada uno con cantidad de tráfico diferente, donde $\text{tipo1} < \text{tipo2} < \text{tipo3}$

- Minuto 0. Inyectar tráfico a la red MPLS experimental desde la fuente1, sin que este sobrepase el límite de congestión (C), inyectando tráfico de tipo 1.
- Minuto 5. Incrementar el tráfico desde la fuente 2 hasta igualar o superar los umbrales definidos como congestión (C), inyectando tráfico de tipo 2.
- Minuto 10. Incrementar nuevamente el tráfico desde la fuente 2 hasta superar los umbrales definidos como congestión (C). inyectando tráfico de tipo 3.
- Minuto 15. Terminar el tráfico de tipo 3 producido por la fuente 2, sin que la cantidad de tráfico total sea menor al umbral de congestión (C).
- Minuto 20. Terminar el tráfico de tipo 2 inyectado por la fuente 2 haciendo que la cantidad de tráfico total sea menor al umbral de congestión (C).
- Minuto 25. Terminar el tráfico total.



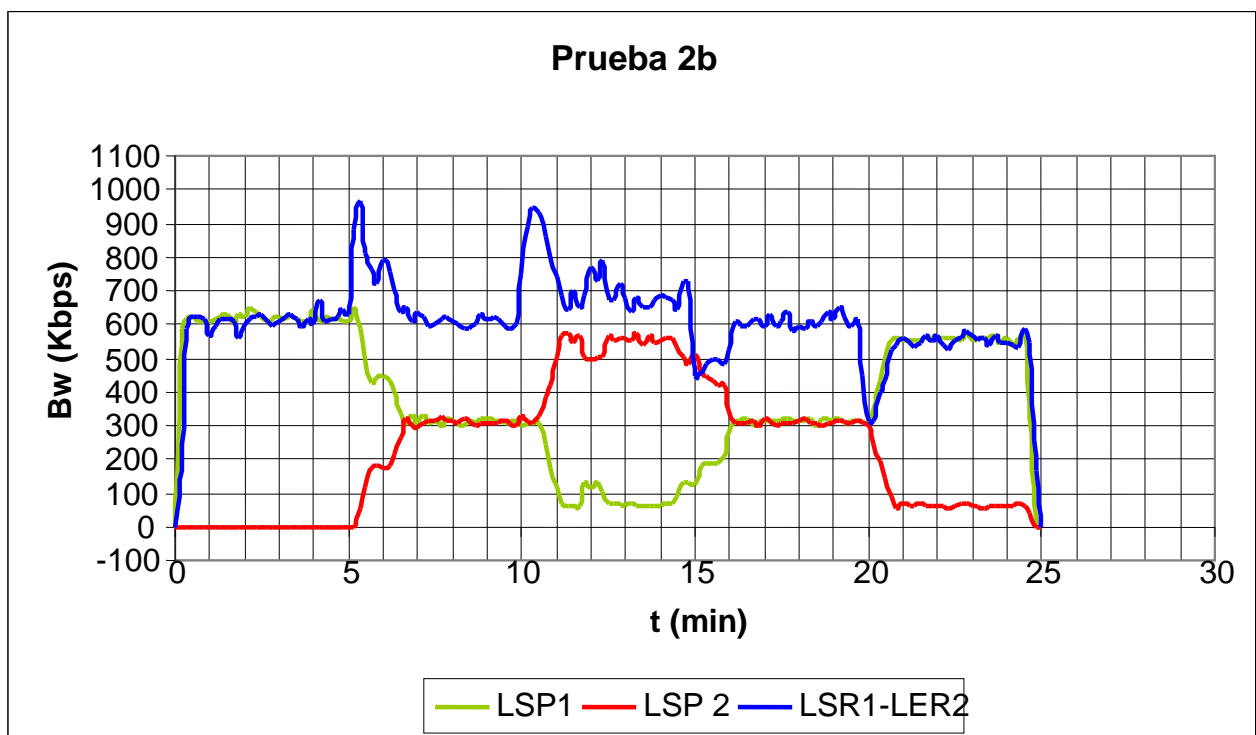
Tabla 7: Tráfico Prueba 2B

<i>Tiempo(mins)</i>	<i>N° Flujos</i>	<i>BW Total</i>	<i>Tráfico LER1</i>	<i>Tráfico LSRI</i>
0	10	600	10	0
5	13	900	0	3
10	15	1200	0	2
15	13	900	0	-2
20	10	600	0	-3
25	0	0	-10	0

El tráfico total de la prueba 2B es similar al tráfico que se envió en la prueba 1B

6.3.3.2.2 Resultados de la Prueba 2B

Figura 24: Resultados Prueba 2B





6.3.3.2.3 Análisis de Resultados, Prueba 2B

Como se puede observar en la figura 24, los resultados obtenidos son similares a los de la prueba 2A, presentando únicamente diferencias en los tiempos utilizados en la detección del estado de congestión que fueron analizados en la prueba 1B.

6.3.3.2.3 Prueba 3B

- Umbral C= 90% y Umbral M= 70%
- Se trabajará con un solo tipo de tráfico.

6.3.3.2.3.1 Cronograma de la Prueba 3A

Se variarán los umbrales C y M, que antes estaban en 70% y 50% incrementándolos a 90% y 70% respectivamente.

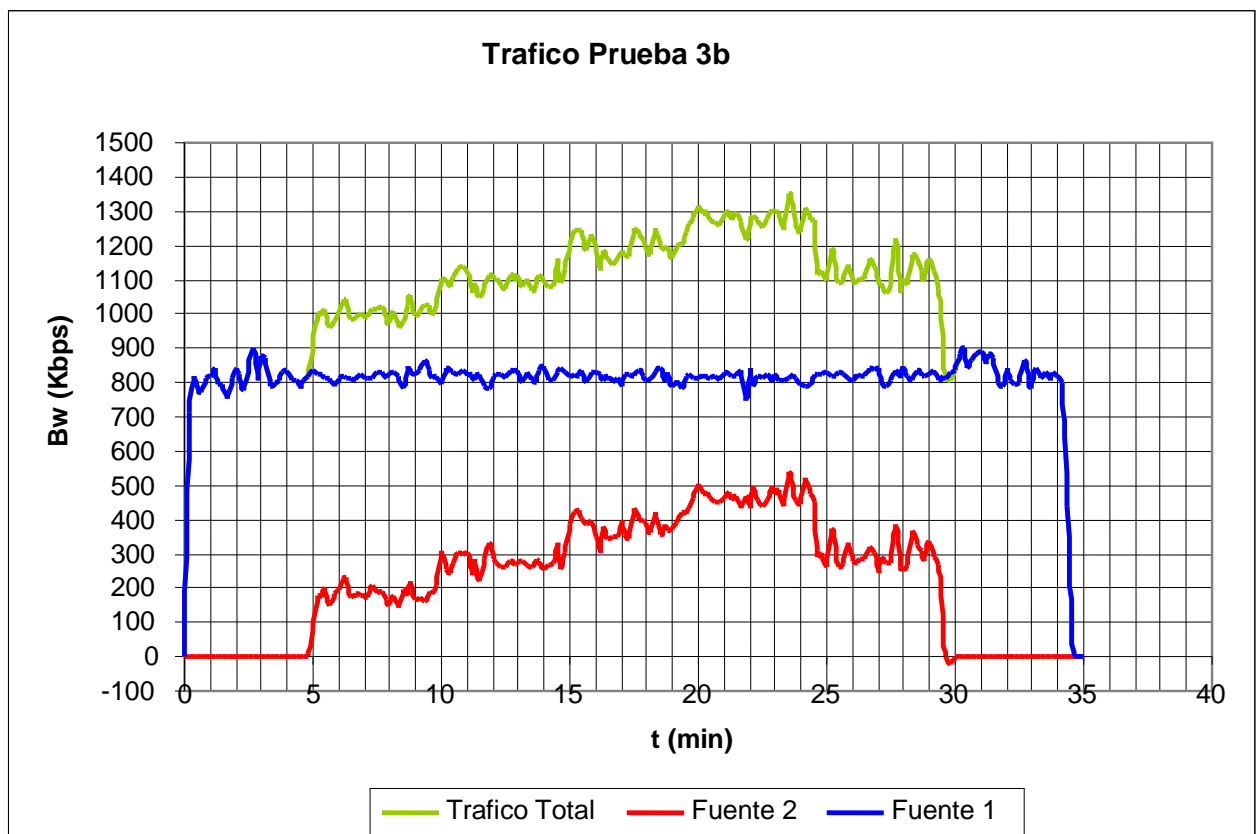
- Minuto 0. Inyectar tráfico a la red MPLS experimental, sin que este sobrepase los límites de congestión (C), desde la fuente 1.
- Minuto 5. Incrementar el tráfico hasta igualar o superar los umbrales definidos como congestión (C), desde la fuente 2.
- Minuto 10, 15 y 20. Incrementar nuevamente el tráfico hasta superar los umbrales definidos como congestión (C), desde la fuente2.
- Minuto 25. Terminar el tráfico de la fuente2 que se inició en el minuto 15 y 20, sin que la cantidad de tráfico total sea menor al umbral de congestión (C).
- Minuto 30. Terminar el tráfico de la fuente2 que se inicio en el minuto 5 y 10.
- Minuto 35 Terminar el tráfico total.



Tabla 8: Tráfico Prueba 3B

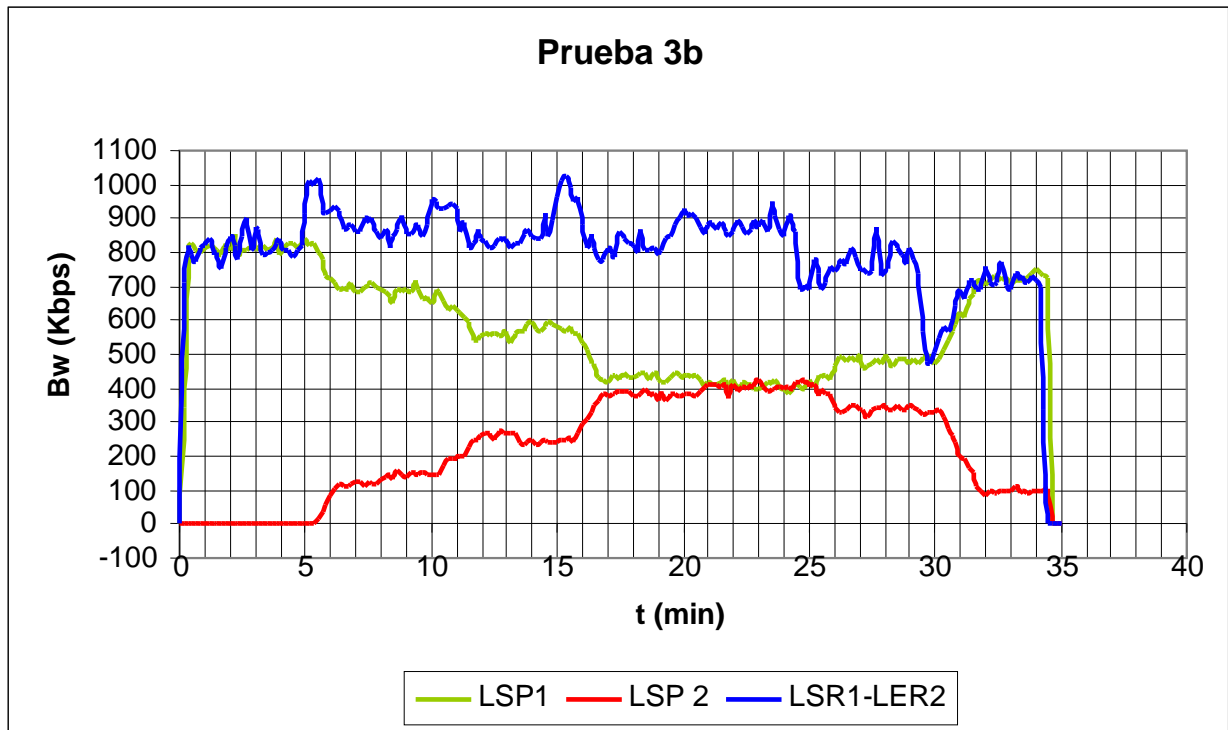
<i>Tiempo(mins)</i>	<i>N° Flujos</i>	<i>BW Total</i>	<i>Tráfico LERI</i>	<i>Tráfico LSRI</i>
0	34	800	34	0
5	42	988	0	8
10	46	1082	0	4
15	50	1176	0	4
20	54	1270	0	4
25	46	1082	0	-8
30	34	800	0	-12
35	0	0	-34	0

Figura 25: Tráfico Total Prueba 3B



6.3.3.2.3.2 Resultados de la Prueba 3B

Figura 26: Resultados Prueba 3B



6.3.3.2.3.3 Análisis de Resultados, Prueba 3B

Como se puede observar en la figura 26, los resultados obtenidos son similares a los de la prueba 3A, presentando únicamente diferencias en los tiempos utilizados en la detección del estado de congestión que fueron analizados en la prueba 1B.

6.4 ANÁLISIS DE RESULTADOS GENERALES

Después de haber realizado las pruebas con las cuales se mostró el desempeño y rendimiento del mecanismo DLCCM, se presenta a continuación un diagrama que resume el comportamiento general de cada uno de los escenarios en que se probó el mecanismo.



6.4.1 Resultados Escenario Nº 1.

La tabla 9 muestra los resultados obtenidos del mecanismo de balanceo de carga cuando el tráfico actual de la red (estado actual de la red) no presenta Cross traffic.

Tabla 9: Resumen de Resultados, sin Cross traffic en la red.

Prueba Nº	Umbral C	Umbral M	Tipos de Tráfico	Tiempo de Respuesta (aproxim.)	Utilización Máxima del Enlace	Precisión del Balanceo
1	70%	50%	1	3 min	68%	Alta
2	70%	50%	1,2,3	1.3 min	64%	Media
3	90%	70%	1	1.5 min	86%	Alta

- **Tiempo de respuesta:** En la prueba 1 el tiempo de respuesta fue relativamente alto debido a que el tipo de tráfico que se utilizó era pequeño, razón explicada en 6.1.2.5. En la prueba 2 el tiempo de respuesta fue de 1.3 min aproximadamente, el cual es menor que el tiempo empleado en la prueba 1, esto fue debido al incremento en el tamaño de los flujos de tráfico. En la prueba 3 el tiempo empleado fue de 1.5 min, este tiempo es relativamente pequeño ya que la cantidad de tráfico a balancear fue menor debido a que los umbrales de congestión se incrementaron.
- **Utilización Máxima del Enlace:** En la prueba 1 la utilización máxima del enlace fue de 68%, esto fue debido a que los flujos de tráfico que ingresaron a la red eran pequeños, ocasionando que el balanceo de carga fuera preciso y que se aproximara al umbral de congestión C. En la prueba 2 la utilización máxima del enlace disminuye, debido a que los flujos de tráfico que ingresaron a la red eran más grandes. En la prueba 3 la utilización máxima del enlace es mucho mayor ya que los umbrales de congestión fueron incrementados notoriamente.
- **Precisión del Balanceo:** En la prueba 1 la precisión del balanceo fue alta, debido a que los flujos de tráfico que entraron a la red eran pequeños, razón explicada en



6.1.2.5. en la prueba 2 la precisión de balanceo disminuyo ya que el tamaño de los flujos de tráfico entrantes aumento. En la prueba 3 la precisión también fue alta debido a que los flujos de tráfico eran pequeños.

6.4.2 Resultados Escenario Nº 2.

La tabla 10 muestra los resultados obtenidos del mecanismo de balanceo de carga cuando el tráfico actual de la red (estado actual de la red) depende de Cross traffic en el core de la red.

Tabla 10: Resumen de Resultados, con Cross traffic en la red.

Prueba Nº	Umbral C	Umbral M	Tipos de Tráfico	Tiempo de Respuesta (aproxim.)	Utilización Máxima del Enlace	Precisión del Balanceo
1	70%	50%	1	3 min	68%	Alta
2	70%	50%	1,2,3	2 min	64%	Media
3	90%	70%	1	1.5 min	86%	Alta

Los resultados obtenidos en el escenario 2 fueron similares a los que se obtuvieron en el escenario 1. Se presentaron cambios únicamente en los tiempos de respuesta que el mecanismo utilizaba para balancear carga, esto se presento debido a los retardos en la detección del estado actual de la red, ya que en este escenario el estado actual de la red dependía del cross traffic presente en el core de la red.



VII. CONCLUSIONES

Respecto al Trabajo Mismo.

- Es posible plantear soluciones de balanceo de carga en MPLS utilizando las extensiones para ingeniería de tráfico de los protocolos OSPF y RSVP, ya que estos protocolos proporcionan en conjunto las características funcionales necesarias en un mecanismo de balanceo de carga.
- Para desarrollar mecanismos de balanceo de carga utilizando los protocolos RSVP-TE y OSPF-TE, se debe integrar las características que cada uno de ellos proporciona, ya que estas características son complementarias y facilitan que las ventajas obtenidas en el plano de control sean aprovechadas desde el plano de reenvío y viceversa. Donde RSVP-TE se utiliza para el establecimiento de caminos explícitos y OSPF como protocolo de estado enlace, para distribuir la información del estado actual de los enlaces en todo el dominio MPLS.
- Se identificaron problemas relevantes en los mecanismos de balanceo de carga tales como: La alta complejidad que pueden presentar los mecanismos que ejecutan un control proactivo, Cómo obtener el estado actual de la red, los mecanismos planteados parten del hecho de conocer el ancho de banda de cada nueva conexión, Cómo clasificar el tráfico entrante que se desea balancear, Cómo asignar dinámicamente el tráfico sobre diferentes rutas establecidas, Cómo determinar la cantidad de tráfico que debe ser movida de un enlace a otro. Estos problemas permitieron determinar las características funcionales que debe tener un mecanismo balanceo de carga, para que sea viable su implementación en un entorno real.
- Acorde a las recomendaciones del RFC 2702, se definieron los siguientes parámetros: Control, Medidas, Caracterización de la red, Integración de los procesos de enrutamiento con los de conmutación, Todos los anteriores a un bajo costo. Los cuales determinan la efectividad y viabilidad de implementación de un mecanismo de balanceo de carga en MPLS.



- Los mecanismos CBR presentan más dificultades a la hora de su implementación que los mecanismos basados en reparto o división de tráfico, ya que presentan altos niveles de complejidad en sus algoritmos de balanceo de carga y a que no tienen en cuenta la carga de tráfico en los enlaces para tomar las decisiones de enrutamiento.
- El balanceo de carga dinámico permite a los mecanismos de balanceo de carga adaptarse al estado actual de la red, siendo este el mayor problema que presentan los mecanismos de balanceo estático.

Respecto a la Tecnología y al Balanceo de Carga

- El balanceo de carga es una opción a la hora de dar solución a los problemas de congestión que presentan las actuales redes de telecomunicaciones, ya que cuando la congestión es minimizada mediante la eficiente asignación de los recursos, la pérdida de paquetes y los retardos de transmisión disminuyen, e incrementa el Throughput total de la red.
- La tecnología MPLS facilita la implementación de mecanismos de Ingeniería de Tráfico como: mecanismos de balanceo de carga para evitar la congestión, utilización óptima de los recursos de la red, actividades de monitoreo que permitan realizar un análisis del desempeño, recuperación ante fallas etc.
- Es posible plantear soluciones que garanticen calidad de servicio a determinados tipos de tráfico, utilizando algunas extensiones del protocolo RSVP-TE.
- Es posible utilizar los LSAs opacos para distribuir información de aplicaciones externas, que contribuyan a mejorar el desempeño de la red.

Respecto a la Simulación y las Herramientas.

- Las medidas en línea de los parámetros que determinan el estado actual de la red hacen que la información obtenida sea una representación aproximada del tráfico



actual, facilitando así que los procesos de balanceo de carga presenten el menor número de errores posibles.

- La utilización de máquinas virtuales es una opción práctica para desarrollar y validar proyectos de investigación relacionados con el networking, cuando los recursos hardware son escasos.
- La utilización de proyectos de software libre permiten realizar investigación y apropiación del conocimiento, ya que la mayor parte del software para equipos de interconexión de redes han sido desarrollados bajo licencias restrictivas que impiden el acceso al código fuente.
- La validación de proyectos de networking bajo implementaciones empíricas, permiten identificar problemas propios de un entorno real tales como: retardos, pérdida de paquetes, caída de los enlaces, altos consumo de CPU y memoria en los nodos, etc. Que no son tenidos en cuenta en una simulación, donde el entorno de desarrollo es ideal.



RECOMENDACIONES Y TRABAJOS FUTUROS

- Se recomienda estar al tanto del estado del arte de los protocolos RSVP-TE y OSPF-TE, y de los proyectos que implementan la tecnología MPLS, para observar que nuevas facilidades se proporcionan a la hora de implementar mecanismos de balanceo de carga.
- Validar el modulo de generalización del algoritmo LCM-Modificado, configurando varios LSPs al mismo tiempo y balanceando carga sobre ellos.
- Adicionar un modulo DiffServ al mecanismo DLCCM, el cual le permita hacer balanceo de carga teniendo en cuenta la calidad de servicio asociada a cada flujo de tráfico, de tal manera que la clasificación del tráfico no se haga por direcciones Ip o puertos de origen y destino si no que se haga clasificando el tráfico con prioridades y acuerdos de nivel de servicio.
- Implementar el mecanismo de balanceo de carga en un lenguaje de programación de alto rendimiento como “C++”, ya que este lenguaje proporciona rapidez de ejecución, y puede ser fácilmente integrado a otros proyectos de networking desarrollados bajo este mismo lenguaje de programación.
- Realizar pruebas al mecanismo DLCCM ejecutándolo sobre máquinas independientes, de tal manera que se pueda determinar la influencia de las máquinas virtuales en los tiempos de respuesta.
- Implementar el mecanismo de balanceo de carga DLCCM con otros protocolos como ISIS y CR-LDP.



REFERENCIAS

- [1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, (Septiembre 1999).
- [2] E. Rosen and A. Viswanathan, "Multiprotocol label switching architecture", RFC 3031, (Enero 2001).
- [3] Albert Statti, "JUNOS Internet Software Configuration Guide: MPLS Applications", Juniper Networks, (Agosto 2001).
- [4] Alcatel, "Traffic Engineering Solutions for Core Networks", Technical Paper, (Junio 2002).
- [5] Pablo Belzarena, "Ingeniería de Tráfico en línea en redes MPLS Aplicando la teoría de grandes desviaciones", Universidad de la Republica Montevideo (Uruguay 2003).
- [6] Miguel Griot Gayoso, Santiago Remersaro Romaniello, Gabriel Tucci Scudroni, "Ingeniería de tráfico en tiempo real sobre MPLS. Implementación en un ambiente Linux". Facultad de Ingeniería. Universidad de la República Montevideo - Uruguay. (Mayo 2003).
- [7] María Sol Canalis, "MPLS Multiprotocol Label Switching: Una Arquitectura de Backbone para la Internet del Siglo XXI", Dpto. Informática. Universidad Nacional del Nordeste- Argentina, (2000).
- [8] Fernando Alcober Bayo, "Curso de Teleeducación Sobre MPLS", Universidad Politécnica de Madrid Facultad de Informática, http://www.fi.upm.es/~jgarcia/curso_MPLS/, (Mayo 2001).
- [9] Adrian Farrel, "The Internet and Its Protocols A Comparative Approach", Morgan Kaufmann, (Sep 2004).



[10] Awduche, D. et al., "Overview and Principles of Internet Traffic Engineering", RFC 3272, (May 2002).

[11] Elio Salvadori, Roberto Battiti, "A Load Balancing Scheme for Congestion Control in MPLS Networks", 2003.

[12] MPLS for Linux. <http://mpls-linux.sourceforge.net>, 2005.

[13] ZebOS - Advanced Routing Suite. <http://www.ipinfusion.com>, 2005.

[14] NIST Switch. <http://www.antd.nist.gov/nistswitch>, Mayo 2000.

[15] D.Awduche, J.Malcolm, J.Agogbua, M.O'Dell, J.McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, Sep 1999.

[16] Xavier Hesselbach, Yezid Donoso, Ramón Fabregat, Fernando Solano, Benjamín Baran, Mónica Huerta, "Hashing based traffic partitioning in a multicast-multipath MPLS network model", 2005.

[17] Oscar J. Calderón C, Xavier Hesselbach S. "Balanceo de Carga en Redes MPLS", Revista Gerencia Tecnológica Informática. Número 5. Volumen 2, 2003.

[18] Keping Long, Zhongshan Zhang, Shiduan Cheng, "Load balancing algorithms in MPLS traffic engineering", Mayo 2001

[19] A. Jüttner, B. Szviatovszki, A. Szentesi, D. Orincsay, J. Harmatos, "On-demand Optimization of Label Switched Paths in MPLS Networks", Oct. 2000.

[20] K. Kar, M. Kodialam, T. Lakshman, "Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications", Dic 2000.

[21] B. Wang, X. Su, and C. Chen. "A New Bandwidth Guaranteed Routing Algorithm for MPLS Traffic Engineering", 2002.



[22] Gao D., Shu Y., Liu S. Yang O.W.W., "Delay-Based adaptive load balancing in MPLS networks", IEEE ICC, May 2002.

[23] Xavier Hesselbach, Mónica Huerta, Oscar Calderón. "Problemas abiertos en MPLS. Migración, Protección, Gestión de Recursos y Balanceo de Carga", 2003

[24] Felicia Marie Holness, "Congestion Control Mechanisms within MPLS Networks", Sep 2000.

[25] J. M. Arco, A. García, E. Castro, J. A. Carral, "LCM, A LOAD BALANCE ALGORITHM IN MPLS-TE", 2005.

[26] David Allan, Guidelines for MPLS load balancing, draft-allan-mpls-loadbal-04, IETF, Abril 2003.

[27] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification", RFC 2205, Septiembre 1997.

[28] D. Awduche, L. Berger, D. Gan ,T. Li, V. Srinivasan, "Extensions to RSVP for LSP Tunnels", RFC 3209, Diciembre 2001.

[29] "Protocolo de enrutamiento OSPF", <http://www.solont.com/z-net/ospf/ospf.htm>, Dic 2000.

[30] "OSPF Tutorial", <http://us.geocities.com/pelicansmith/work/ospf.html>, Dic, 2000.

[31] "OSPF Design Guide", <http://www.cisco.com/warp/public/104/1.html>, Abr 2006.

[32] J. Moy,"OSPF Version 2", RFC 2328, Abr 1998.

[33] R. Coltun, "The OSPF Opaque LSA Option", RFC 2370, Jul 1998.

[34] D. Katz, K. Kompella, D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, Sep 2003.



[35] “An extended Quagga/Zebra OSPF daemon supporting an API for external applications”, <http://www.tik.ee.ethz.ch/~keller/ospfapi/>.

[36] Bourne Again Shell. <http://www.gnu.org/software/bash>, Noviembre 2003.

[37] Quagga Routing Suite, <http://www.quagga.net>, Agosto 2006.

[38] RSVP-TE daemon for DiffServ over MPLS under Linux, <http://dsmpis.atlantis.ugent.be>, Mayo 2004.

[39] Netfilter. <http://www.netfilter.org>, Agosto 2006.

[40] VMware - Virtualization Software. <http://www.vmware.com>, Agosto 2006.

[41] David C. Reeve, A New Blueprint for Network QoS.
<http://dreeve.org/thesis/thesis.html>, Agosto 2003.

[42] MGEN The Multi-Generator Toolset, <http://pf.itd.nrl.navy.mil/mgen>, 2006.