

Framework para el desarrollo del encaminamiento interdominio con QoS basado en PCE sobre MPLS

UNIVERSIDAD DE EXTREMADURA

GÍTACA

Grupo de investigación de
Ingeniería Telemática Aplicada y Comunicaciones Avanzadas

www.gitaca.es

Agosto de 2008

Autores: Jaime Galán Jiménez – Miguel Ángel Melón Pérez

Tutor: José Luis González Sánchez



Framework para el desarrollo del encaminamiento interdominio con QoS basado en PCE sobre MPLS

Autores: Jaime Galán Jiménez y Miguel Ángel Melón Pérez
Director del trabajo: José Luis González Sánchez

19 de octubre de 2008

Índice

I	Estudio previo de las tecnologías	6
1.	Encaminamiento	6
1.1.	Encaminamiento estático o determinístico	6
1.2.	Encaminamiento dinámico o adaptativo	6
1.2.1.	Encaminamiento por inundación (flooding)	7
1.2.2.	Vector distancias	7
1.2.3.	Estado del enlace	8
2.	Conmutación de etiquetas	10
2.1.	La clase de equivalencia funcional: FEC	10
2.1.1.	Escalabilidad y grado de granulado	11
2.2.	Funciones de control y reenvío	11
2.3.	Alternativas para el transporte de la etiqueta	12
2.4.	La tabla de encaminamiento	12
2.5.	Etiquetas libres	13
2.6.	Asociación de etiquetas a las FEC	13
2.6.1.	Asociación local y asociación remota	13
2.6.2.	Asociación río arriba (Upstream) y asociación río abajo (Downstream)	13
2.6.3.	Asociación de etiquetas dirigida por control o por datos	14
2.7.	Label swapping: Intercambio de etiquetas	14
3.	MPLS	15
3.1.	Terminología	16
3.2.	Tipos de nodos MPLS	17
3.3.	Protocolos de distribución de etiquetas	18
3.3.1.	Distribución y asignación de etiquetas	18
3.3.2.	Formato de las etiquetas	19
3.3.3.	La pila de etiquetas	20
3.4.	Aplicaciones de MPLS	21
3.4.1.	Ingeniería de tráfico	21
3.4.2.	Soporte a las clases de servicio (CoS)	22
3.5.	MPLS-TE	23
3.6.	GMPLS	23

4. BGP	24
4.1. Evolución histórica de BGP	24
4.2. Funcionamiento de BGP-4	24
4.3. MPLS-BGP	27
4.4. Situaciones de BGP	28
4.4.1. Anuncio de múltiples rutas a un destino	28
4.4.2. Hermanos BGP que no son adyacentes	28
4.5. Problemática de BGP	28
5. RSVP	29
5.1. Características de RSVP	29
5.2. Flujos de datos	30
5.3. Mensajes RSVP	30
5.3.1. Formato de la cabecera	31
5.3.2. Campos de la cabecera	31
5.3.3. Formato de los objetos	31
5.3.4. Funcionamiento de RSVP	32
5.4. RSVP-TE. Extensiones de RSVP para túneles LSP	32
6. PCE	34
6.1. Funcionamiento de PCE	35
II Simulaciones e investigación de cuestiones propuestas	37
7. Búsqueda de entornos de simulación	37
7.1. OPNET Modeler	38
7.1.1. Estudio comparativo de OPNET Modeler	38
7.2. TOTEM Project	39
7.2.1. Características disponibles en TOTEM	40
8. Colaboración con el proyecto OpenSimRIPCA	42
8.1. Validación XML	42
8.1.1. Parsers XML	42
8.2. API's de XML para Java	43
8.2.1. DOM	43
8.2.2. SAX	44
8.3. Aportación a OpenSimRIPCA	44

9. Relaciones entre OSPF-TE, BGP y PCE	44
9.1. OSPF-TE y PCE	45
9.2. BGP y PCE	45
9.2.1. PCE Discovery Attribute	45
9.2.2. Funcionamiento del PCE Discovery Attribute	46
10. Conclusiones y trabajo futuro	46
III Anexos	47

Introducción

En el presente documento se recoge, a modo de memorándum, el trabajo realizado y las impresiones y conocimientos obtenidos durante el desarrollo de la beca *Framework para el desarrollo del encaminamiento interdominio con QoS basado en PCE sobre MPLS*, proyecto realizado en convenio entre la empresa CISCO Systems, la Junta de Extremadura y la Universidad de Extremadura, a la que pertenece el grupo GÍTACA. La duración inicial de dicho proyecto es de un año, concluyendo el 15 de septiembre de 2008.

Para poder hacer frente a las propuestas establecidas en la beca, inicialmente ha habido que formarse en todas las tecnologías relacionadas con la misma para poder tener un nivel aceptable que permitiese adentrarnos en dichas propuestas. Estos conceptos son los siguientes:

- Nociones básicas sobre encaminamiento (*routing*).
- MPLS. MPLS-TE (*Multiprotocol Label Switching – Traffic Engineering*).
- GMPLS (*Generalized Multiprotocol Label Switching*).
- RSVP. RSVP-TE (*Resource Reservation Protocol – Traffic Engineering*).
- BGP-4 (*Border Gateway Protocol – versión 4*).
- OSPF. OSPF-TE (*Open Shortest Path First – Traffic Engineering*).
- PCE (*Path Computation Element*).

Seguidamente, y una vez adquiridos los conocimientos supuestos para el desarrollo de este proyecto, se han encaminado los esfuerzos hacia la realización de simulaciones sobre determinados escenarios para poner a prueba las capacidades de la arquitectura PCE. De esta manera, se podrá apoyar con datos tangibles los resultados obtenidos de las investigaciones enmarcadas dentro de los objetivos principales de esta beca.

A continuación se van a explicar de forma resumida cada uno de los puntos estudiados y nombrados anteriormente.

Parte I

Estudio previo de las tecnologías

1. Encaminamiento

El encaminamiento es el mecanismo empleado en una red de interconexión para determinar la ruta óptima hacia un destino. Existen dos tipos de encaminamiento claramente diferenciados que se explican en los siguientes puntos:

- Encaminamiento estático o determinístico.
- Encaminamiento dinámico o adaptativo.

1.1. Encaminamiento estático o determinístico

En este tipo de encaminamiento, la información que debe usarse para alcanzar el nodo destino se almacena en las tablas de encaminamiento de los nodos cuando éstos se ponen en funcionamiento y no se tiene en cuenta el estado de la subred a la hora de tomar las decisiones de encaminamiento.

En caso de existir múltiples caminos hacia el destino se suele utilizar una segunda métrica, como por ejemplo la distancia, y si las tablas de encaminamiento se configuran de forma manual, deben estar basada en documentación de red. Éstas, además, permanecen inalterables hasta que no se vuelva a actuar sobre ellas; por tanto, la adaptación en tiempo real a los cambios de las condiciones de la red es nula.

El cálculo de la ruta óptima es también off-line, por lo que no importa ni la complejidad del algoritmo ni el tiempo requerido para su convergencia. Aunque estos algoritmos son rígidos, rápidos y de diseño simple, son los que peores decisiones toman en general.

Existe una desventaja clara en este tipo de encaminamiento, que es la posibilidad que se puedan producir cambios en la topología o fallos, lo que implica modificar las tablas de encaminamiento de todos los routers, de forma que es inapropiado para redes grandes o con cambios frecuentes, como Internet.

Entre las ventajas que presenta el encaminamiento estático, destaca el hecho de que no es necesario el despliegue de protocolos de encaminamiento, ya que se produce un despliegue mucho más sencillo al tratarse de redes pequeñas.

Por el contrario, este tipo de encaminamiento es inadecuado para organizaciones con un diseño de red complejo, para redes con continuos cambios topológicos y para redes tolerantes a fallos o que transporten tráfico difícilmente predecible.

1.2. Encaminamiento dinámico o adaptativo

Dentro del encaminamiento dinámico o adaptativo existen dos familias, que son las siguientes:

- *Interior Gateway Protocol* (IGP) o protocolo de encaminamiento interior.
- *Exterior Gateway Protocol* (EGP) o protocolo de encaminamiento exterior.

En este punto, se debe explicar el concepto de Sistema Autónomo (*Autonomous System* o AS):

Conjunto de redes IP y routers bajo una misma autoridad administrativa que presenta una misma política de encaminamiento hacia Internet.

Así pues, el IGP está definido por la autoridad administrativa del AS (encaminamiento intradominio) y el EGP es común para conseguir el encaminamiento interdominio, entre AS.

1.2.1. Encaminamiento por inundación (flooding)

En el encaminamiento por inundación, cuando un nodo recibe un paquete, reenvía una copia a cada uno de los nodos con los que está conectado, de forma que la información suele alcanzar el destino a través de la ruta óptima, pero a costa de una fuerte sobrecarga. Este tipo de encaminamiento se usa solamente en las fases de inicialización, para que los routers conozcan la topología de la red.

1.2.2. Vector distancias

Se trata de un algoritmo distribuido empleado por los routers para construir una tabla de encaminamiento denominada vector que contiene el coste de cada camino, es decir, la distancia para alcanzar cualquier destino.

Inicialmente, un router solamente conoce a los nodos a los que está directamente conectado, y para construir las tablas de encaminamiento completas, cada router envía en intervalos de tiempo predefinidos su tabla actualizada con los vecinos y las distancias que conoce.

Este algoritmo es más eficiente que el encaminamiento por inundación porque solamente se emplea una ruta para intentar llegar al destino, pero las actualizaciones implican un *overhead* y coste de procesamiento en los nodos. Además, los paquetes pueden entrar en bucles en lugar de ir directamente al destino, con el problema que esto supone.

Un ejemplo de protocolo que utiliza este tipo de algoritmo es RIP, que mostramos a continuación.

RIP

RIP (*Routing Information Protocol*) es un protocolo IGP simple basado en el algoritmo vector distancias [1]. El primer desarrollo de RIP fue un componente del código de red de Berkeley UNIX, denominado *routed (route management daemon)*.

Seguidamente se desarrollaron dos nuevos tipos: RIPv1 y RIPv2, que pasamos a explicar a continuación.

RIPv1

RIPv1 se desarrolló para requerir una cantidad de configuración mínima y una complejidad de desarrollo pequeña para facilitar su despliegue. Esto se realizó debido a la simplicidad del protocolo.

Las características de este protocolo son las siguientes:

- Las direcciones que se encuentran en las tablas RIP son direcciones IP de 32 bits.
- Una entrada en las tablas de routing puede representar un host, una red o una subred.
- Inicialmente se separa la parte de dirección de red de la parte “subred+host” como una función dependiente de la clase de red.
- Si el par “subred+host” es nulo, la dirección representa a una red.
- Si por el contrario dicho par no es nulo, la dirección representa a una subred o un host.
- Como métrica empleada, utiliza el hop-count, con un número máximo de 15 saltos.

Existen dos mecanismos implementados en RIP para evitar que se pueda propagar información de routing incorrecta; éstos son Holddown y Split horizon, donde el primero de ellos se establece a 180 segundos.

Por último, decir que las actualizaciones de las tablas de encaminamiento completas se envían por defecto cada 30 segundos mediante broadcasting.

RIPv2

Esta versión de RIP [2] da soporte a CIDR, incorpora la autenticación, que, aunque inicialmente estaba basada en texto plano, posteriormente se incorporó MD5. Las actualizaciones se envían a RIP2-ROUTERS-MCAST.NET (224.0.0.9).

Después de haber visto las dos versiones de RIP, hay que decir que es un protocolo simple, sencillo de desplegar y con una configuración mínima. En cuanto a los inconvenientes, RIP es inadecuado para redes grandes y complejas y, aunque puede calcular rutas nuevas en caso de que se produjeran cambios en la topología de red, en algunos casos lo hace muy lentamente. Cuando esto sucede, la red queda en un estado transitorio. Está limitado a 15 hops y podrían ocurrir ciclos en la red y causar congestión.

1.2.3. Estado del enlace

Se trata de un algoritmo que intenta encontrar siempre el camino más corto, es decir, su objetivo es elegir el camino más corto desde cualquier nodo de la topología de red hacia el resto, utilizando para ello Dijkstra.

En un principio, cada router solamente conoce su propia tabla de conectividad o adyacencia, no la de los demás. Periódicamente, cada nodo difunde a sus vecinos inmediatos un mensaje de estado de los enlaces, con cada router que conoce y su información de conectividad asociada. Si no existiera conectividad entre dos nodos, la distancia entre ambos se considera infinita.

Si se produjese un empate entre dos o más rutas, se elige arbitrariamente una, pero se conservan todas en la tabla, de forma que se haga posible la compartición de la carga entre todas ellas (Ingeniería de Tráfico). Además de la distancia, se pueden asociar otros costes o restricciones a los enlaces (ancho de banda disponible, retardo, etc.).

Entre los protocolos de encaminamiento de estado de enlace se encuentran OSPF e IS-IS.

OSPF

OSPF (*Open Shortest Path First*) [3] se trata del sucesor natural de RIP; se desarrolla en la tecnología de estado de enlace con objeto de solucionar algunos de los problemas del algoritmo vector distancia y es la actual recomendación del IAB (*Internet Architecture Board*).

En este caso, en lugar de intercambiar distancias a cada destino, se mantiene un mapa de la red que se actualiza rápidamente tras cada cambio en la topología de la misma. Para ello, dada una base de datos de estado de enlace, se puede utilizar cualquier algoritmo para el cálculo de rutas.

OSPF utiliza el algoritmo SPF y presenta tres versiones: OSPFv1, OSPFv2 y OSPFv3. Entre las características que presenta este protocolo de encaminamiento interior, se pueden citar las siguientes:

- Convergencia más rápida y sin crear ciclos.
- Soporte de métricas múltiples.
- Soporte de varias rutas a un mismo destino.
- Representación independiente para las rutas externas.
- La métrica en la que se basa es el path cost.
- Es jerárquico, es decir, permite la posibilidad de dividir un Sistema Autónomo en varias áreas.
- Utiliza IP directamente, es decir, no usa ni TCP ni UDP.
- Los subprotocolos que presenta son Hello Protocol, Exchange Protocol y Flooding Protocol.

OSPF-TE

OSPF-TE¹ [4] es un protocolo de encaminamiento intradominio o interior jerárquico, basado en OSPF pero con la extensión de ingeniería de tráfico. Utiliza el algoritmo SPF para calcular la ruta más corta posible, siendo el coste de mandar los paquetes por dicho enlace su medida métrica. Además, construye una base de datos con el estado de cada enlace de la red llamada *Link State Database* (LSD), idéntica en todos los enrutadores de la zona.

Dentro de un *Sistema Autónomo* (AS), un ISP está capacitado para ofrecer cierta garantía de servicio a sus clientes puesto que todo lo que tiene lugar en él queda dentro de sus redes. Para ello, se ayuda de protocolos de encaminamiento interior como OSPF-TE (*Open Shortest Path First-Traffic Engineering*), similar al protocolo OSPF con ingeniería de tráfico.

Cuando un AS utiliza OSPF-TE como protocolo de encaminamiento interior, se organiza internamente de una forma concreta que permite que sea eficiente y que funcione de una forma apropiada.

Inicialmente se subdivide en áreas, donde existen nodos que se denominan *Interior Router* (IR). Todas las áreas se encuentran comunicadas entre sí a través de una troncal (que es a su vez otro área) mediante nodos llamados *Area Border Router* (ABR). De este modo, todas las áreas están comunicadas, pero se contiene el flujo de tráfico dentro de cada área mientras no deba “salir a otras áreas”. Eventualmente, pueden existir encaminadores situados en el borde del AS encargados de tramitar todas aquellas conexiones que deban exceder los límites del mismo y que se denominan *AS Border Router* (ASBR). Los ASBR pueden estar directamente conectados a un área o al área que hace de troncal.

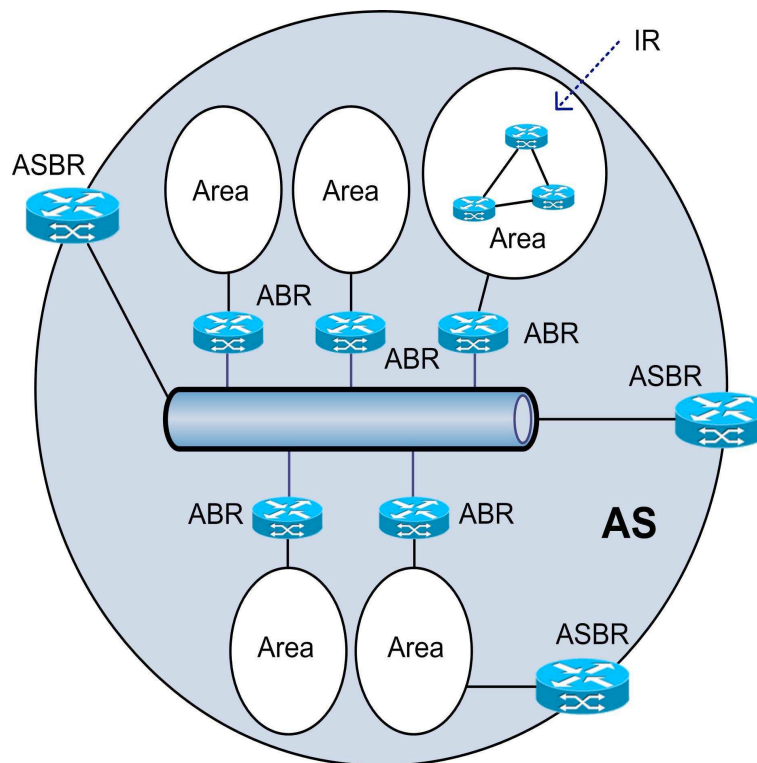


Figura 1: Sistema autónomo usando OSPF-TE

En el anterior esquema, OSPF-TE funciona de la siguiente forma:

- Cada encaminador OSPF-TE ejecuta una máquina de estados finitos del protocolo que es independiente. Se cambia de estado cuando se producen eventos (caída de un enlace, una interfaz, etc.).

¹OSPF-TE está definido en el Request For Comments (RFC) 3630 de la Internet Engineering Task Force (IETF). Se puede consultar en la siguiente dirección web: <http://www.ietf.org/rfc/rfc3630.txt?number=3630>

- Cada encaminador OSPF-TE tiene una base de datos denominada LSD (*Link State Database*) en la que se refleja la topología completa de la red. La LSD es la misma para todos los nodos y éstos se encargan de mantenerlas sincronizadas mediante el intercambio de información sobre el estado de los enlaces de forma frecuente. La LSD contiene información sobre las métricas de los enlaces, las situaciones de fallo de enlaces o nodos, etc.
- Cada encaminador OSPF-TE calcula, sobre la LSD, las rutas más factibles respecto a las restricciones que se establezcan desde él hasta cualquier punto del AS. De este modo, cuando llegue información al nodo, éste sabrá la mejor ruta que se debe seguir hacia el destino.
- El proceso se repite continuamente, tanto el intercambio de información para sincronizar la LSD como el cálculo de las rutas a todos los destinos a partir de ésta, lo que hace que ante fallos eventuales de la red, y al calcular las rutas de nuevo, éstas no tengan en cuenta el tramo que no funciona.

OSPF-TE es el protocolo de encaminamiento interior más utilizado, aunque por sí solo es válido exclusivamente para comunicaciones intradominio.

IS-IS

Tanto RIP como OSPF (OSPF-TE también) son propuestas de protocolos de encaminamiento intradominio (IGP) del IETF, en cambio, OSI tiene la suya propia, que es IS-IS (*Intra-Domain Intermediate System to Intermediate System Routing Protocol*) [5].

Este protocolo, al igual que OSPF, es un protocolo de estado de enlace y si los comparamos entre sí, aunque los conceptos son similares, los subprotocolos son distintos. Además, IS-IS no encamina de forma nativa paquetes IP y, aunque OSPF ha sido extendido por su popularidad, IS-IS generalmente escala mejor en redes grandes. Por otro lado, OSPF soporta hasta 50 routers en un área, a diferencia de IS-IS, que soporta hasta 1000.

CSPF

CSPF (*Constrained Shortest Path First*) [6] es un protocolo de encaminamiento en el que la determinación de las rutas no está diseñada para encontrar la mejor ruta hacia los demás nodos, sino sólo hasta el nodo final de un LSP.

2. Conmutación de etiquetas

Antes de continuar, conviene aclarar un par de términos. Los encaminadores utilizan dos protocolos para retransmitir el tráfico hacia el receptor: uno retransmite los paquetes hacia su destino y el otro se encarga de encontrar un camino para que los paquetes puedan viajar hacia su destino.

El primero de estos protocolos se llamaba antiguamente encaminamiento y el segundo descubrimiento de la ruta. Actualmente, el término encaminamiento se usa para referirse al segundo protocolo, y los términos reenvío y conmutación para referirse al primero.

2.1. La clase de equivalencia funcional: FEC

La clase de equivalencia funcional (FEC, *Functional Equivalent Class*) se usa para describir una asociación de paquetes a una dirección destino, o lo que es lo mismo, un grupo de paquetes IP que se reenvían de la misma manera (por el mismo camino, con el mismo tratamiento en el reenvío, etc.) [7]. También se puede asociar el valor de la FEC a una dirección destino y a una clase de tráfico. La clase de tráfico está asociada habitualmente a un número de puerto destino.

Uno de los motivos por los que se utiliza la FEC es porque permite agrupar paquetes en clases. Gracias a esta agrupación, el valor de la FEC en el paquete se puede utilizar para establecer prioridades, de tal forma que se

da más prioridad a unas FEC que a otras. Se pueden usar las FEC para dar soporte a operaciones eficientes de QoS; por ejemplo: se pueden asociar FEC de alta prioridad a tráfico de voz en tiempo real, de baja prioridad a correo, etc.

MPLS, para establecer la relación entre una FEC y un paquete, usa la etiqueta. Dicha etiqueta identifica una FEC específica. Para diferentes clases de servicio se utilizarán diferentes FEC y sus correspondientes etiquetas asociadas.

Una parte esencial de la tabla de encaminamiento mantenida por un determinado encaminador es la dirección del siguiente encaminador. Un paquete perteneciente a una FEC asociado a una determinada entrada de la tabla se reenviará al siguiente encaminador según esté especificado en dicha tabla.

2.1.1. Escalabilidad y grado de granulado

Un aspecto importante de una FEC es su grado de granulado. Si consideráramos una FEC en la que se incluyeran todos los paquetes en los que la dirección destino del nivel de red coincidiera con un determinado prefijo de dirección, tendríamos un granulado grueso. Como contrapartida, el sistema sería muy escalable. El inconveniente es que con un granulado grueso no podríamos diferenciar diferentes tipos de tráfico y por tanto no permitiría clases de tráfico ni operaciones de QoS.

En el otro extremo tendríamos el granulado fino, en el que una FEC podría incluir sólo los paquetes pertenecientes a una aplicación entre dos ordenadores, es decir, paquetes que tengan las mismas direcciones origen y destino, los mismos puertos e incluso la misma clase de servicio. En este caso tendríamos más clasificaciones de tráfico, más FEC, más etiquetas y una tabla de encaminamiento más grande. En consecuencia, una red de conmutación de etiquetas permite distintos grados de granulado de la FEC.

2.2. Funciones de control y reenvío

Podemos distinguir claramente dos componentes: el componente de control y el componente de reenvío, los cuales se explican a continuación.

El componente de control utiliza los protocolos estándar de encaminamiento, como OSPF e IS-IS, que intercambian información de encaminamiento con los encaminadores para construir y mantener las tablas de encaminamiento. Además, el componente de control debe crear las asociaciones entre etiquetas y FEC y distribuir esta información.



Figura 2: Esquema de las funciones de reenvío y encaminamiento

El componente de reenvío envía los paquetes desde la entrada hacia la salida. Para reenviar los paquetes, examina la información de la cabecera del paquete, busca en la tabla de encaminamiento la entrada correspondiente y

reenvía el paquete. Por tanto, el componente de reenvío consiste en el conjunto de procedimientos que usa el encaminador para tomar la decisión sobre el reenvío de un determinado paquete. Estos algoritmos definen la información del paquete que utiliza el encaminador para encontrar una entrada en la tabla de encaminamiento, así como los procedimientos exactos que el encaminador utiliza para encontrar la entrada.

Cada encaminador de la red implementa ambos componentes. Podríamos ver el encaminamiento del nivel de red como una composición de los dos componentes (control y reenvío) implementada de una manera distribuida por el conjunto de encaminadores que conforman la red, y la ventaja fundamental de separar ambos componentes es la posibilidad de modificar uno de ellos sin modificar el otro.

2.3. Alternativas para el transporte de la etiqueta

La decisión del reenvío se basa en uno o más campos del paquete:

Cabecera del nivel de enlace	Cabecera shim	Cabecera del nivel de red (IP)	Cabecera del nivel de transporte	Datos de usuario	Cola del nivel de enlace
------------------------------	---------------	--------------------------------	----------------------------------	------------------	--------------------------

Figura 3: Formato del paquete

El protocolo IP y los números de puerto se usan en la FEC y en las decisiones de reenvío. Estos campos identifican el tipo de tráfico que reside en la parte de datos del datagrama IP, por lo que son muy importantes en redes que admiten diferentes servicios de QoS para diferentes tipos de tráfico.

ATM y Frame Relay (tecnologías del nivel de enlace) llevan la etiqueta en la cabecera del paquete. ATM puede llevar la etiqueta en el campo VCI o en el VPI de la cabecera, mientras que en Frame Relay estará en el campo DLCI de la cabecera. En DWDM la etiqueta puede asociarse con una longitud de onda en la fibra.

Si esta fuera la única opción, en tecnologías como Ethernet que no disponen de un campo para poder llevar la etiqueta en la cabecera del nivel de enlace, no se podría emplear la conmutación de etiquetas. La solución a este problema consiste en transportar la etiqueta en un campo específico para ella, que se inserta entre la cabecera del nivel de enlace y la cabecera del nivel de red. Esta cabecera se denomina *shim header* (cabecera de relleno), que se sitúa en una posición donde la mayoría de los encaminadores pueden procesarla por software, por lo que los encaminadores convencionales pueden convertirse en LSR siempre que tengan el software apropiado. De este modo se permite cualquier tecnología o combinación de tecnologías del nivel de enlace, como por ejemplo la conmutación de etiquetas en redes Ethernet. Por otro lado, el hecho de llevar la etiqueta en el campo VCI de las células ATM permite que un conmutador ATM funcione como un LSR siempre que tenga el software de control apropiado.

2.4. La tabla de encaminamiento

Una tabla de encaminamiento está constituida por múltiples entradas, las cuales constan de los siguientes elementos:

- Etiqueta de entrada
- Una o más subentradas:
 - Etiqueta de salida
 - Interfaz de salida
 - Dirección del siguiente salto

Etiqueta de entrada	Etiqueta de salida Interfaz de salida Dirección del próximo salto	Etiqueta de salida Interfaz de salida Dirección del próximo salto	...
---------------------	---	---	-----

Figura 4: Entradas de la tabla de encaminamiento

Puede haber más de una subentrada, puesto que hay que tratar los paquetes de difusión. De esta forma, se puede enviar un paquete por múltiples interfaces de salida y puede existir una tabla de encaminamiento única o por interfaz, en cuyo caso a la hora de encaminar un paquete habrá que saber la interfaz por donde ha entrado dicho paquete.

La tabla, mantenida por el LSR en cuestión, está indexada por el valor de la etiqueta, de tal forma que la búsqueda en ella es inmediata: únicamente se necesita un solo acceso a memoria, lo que se traduce en un acceso rápido.

2.5. Etiquetas libres

Un LSR mantiene un repositorio de etiquetas libres. Cuando se inicializa un LSR, el repositorio contiene todas las etiquetas que el LSR puede usar para asociaciones locales. En el momento que se crea una asociación, coge una etiqueta del repositorio y cuando la destruye la vuelve a depositar en el repositorio. Si el LSR tiene una tabla de encaminamiento por interfaz, tendrá que tener un repositorio con etiquetas por interfaz.

2.6. Asociación de etiquetas a las FEC

2.6.1. Asociación local y asociación remota

Asociación local. El encaminador local establece la asociación entre la etiqueta y la FEC. Por tanto, la etiqueta pertenecerá al encaminador.

Asociación remota. Un encaminador vecino será el que establezca la asociación, por lo que el encaminador local recibirá la asociación de la etiqueta.

El componente de control usa ambos tipos de asociaciones para poblar su tabla de encaminamiento con etiquetas de entrada y salida.

2.6.2. Asociación río arriba (Upstream) y asociación río abajo (Downstream)

Asociación de etiquetas río abajo (Downstream)

La asociación de la etiqueta a la FEC la realiza el encaminador que está río abajo (Rd) respecto al flujo de paquetes. Por tanto, en la tabla de encaminamiento de Ru tendremos como etiquetas de salida las etiquetas de la asociación remota (puesto que las ha elegido el encaminador que está río abajo) y como etiquetas de entrada las de la asociación local [7].

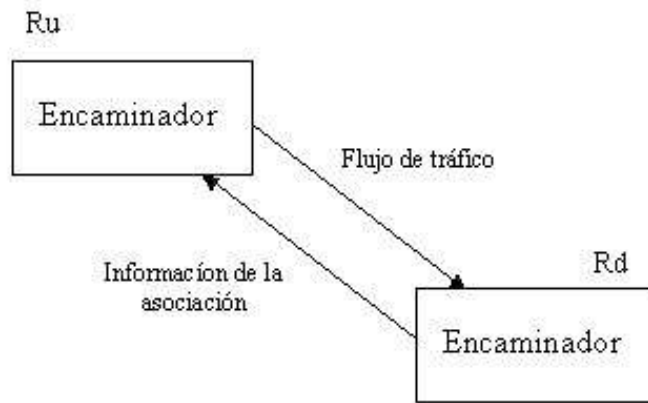


Figura 5: Asociación de etiquetas río abajo

En el ejemplo anterior, Ru le manda un paquete a Rd . Este paquete habrá sido identificado con anterioridad como perteneciente a una FEC y tendrá una etiqueta (E) asociada a ese FEC. Por tanto, Ru le habrá puesto al paquete como etiqueta de salida E .

Asociación de etiquetas río arriba (Upstream)

La asociación de la etiqueta a la FEC la realiza el encaminador que está río arriba (Ru) respecto al flujo de paquetes. Por tanto, en la tabla de encaminamiento de Ru tendremos como etiquetas de salida las etiquetas de la asociación local (puesto que es éste encaminador el que las ha elegido) y como etiquetas de entrada las de la asociación remota.

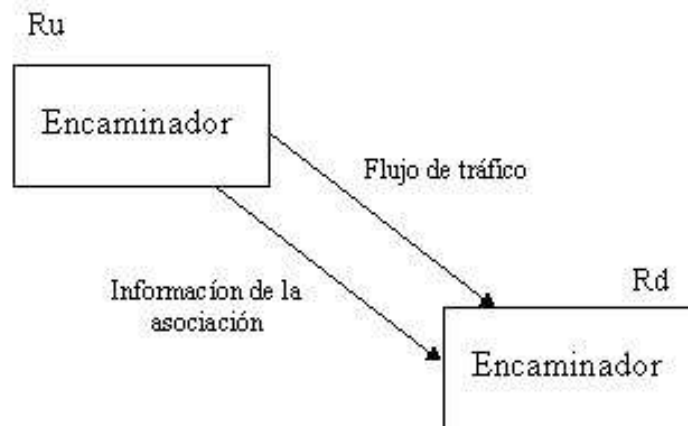


Figura 6: Asociación de etiquetas río arriba

2.6.3. Asociación de etiquetas dirigida por control o por datos

Un LSR crea o destruye asociaciones entre etiquetas y FEC en respuesta a un evento, el cual puede deberse a que recibe información de control o a que debe reenviar paquetes. La asociación de etiquetas a FEC dirigida por control se establece de antemano, y la dirigida por los datos ocurre dinámicamente a medida que fluyen los paquetes. Normalmente, ambos tipos de asociaciones se usan conjuntamente.

2.7. Label swapping: Intercambio de etiquetas

El algoritmo empleado por el componente de reenvío se basa en el intercambio de etiquetas. Cuando un LSR recibe un paquete, extrae el valor de la etiqueta y accede con él a la tabla de encaminamiento, donde encontrará

el nuevo valor de la etiqueta que ha de ponerle al paquete antes de reenviarlo, así como la interfaz de salida por donde ha de mandarlo. También podrá encontrar información sobre si debe o no encolar el mensaje.

El algoritmo de reenvío se suele implementar en hardware por su sencillez. Esto repercute favorablemente en el rendimiento del LSR [8]. Veamos un ejemplo:

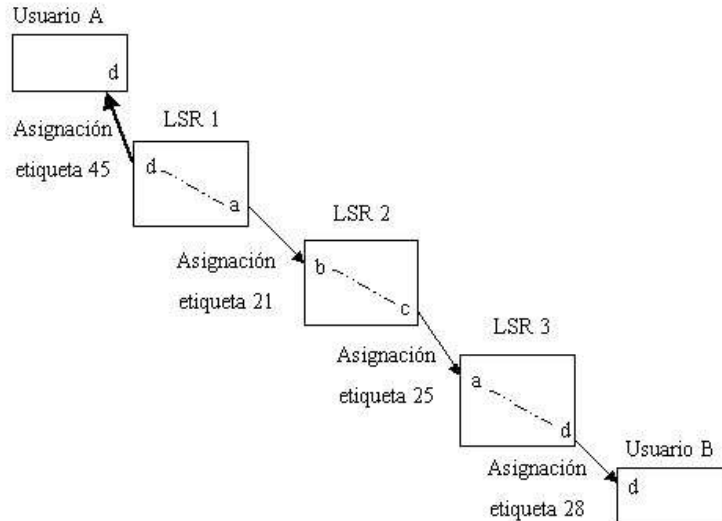


Figura 7: Algoritmo de reenvío

En la figura anterior vemos que:

- La etiqueta 45 identifica el LSP entre el usuario A y el LSR 1
- La etiqueta 21 identifica el LSP entre el LSR 1 y el LSR 2
- La etiqueta 25 identifica el LSP entre el LSR 2 y el LSR 3
- La etiqueta 28 identifica el LSP entre el LSR 3 y el usuario B

3. MPLS

MPLS es un esquema de envío de paquetes que no está formado exclusivamente por un protocolo que encapsula a otros sino que define y utiliza conceptos como LER, LSR, FEC, dominio MPLS, LDP, ingeniería de tráfico, etc. como veremos a continuación.

MPLS se encuentra situado entre los niveles de enlace y de red del modelo de referencia OSI, por tanto, se podría decir que es un protocolo de nivel 2+. Esto, a efectos prácticos, significa que hace de nexo de unión entre los protocolos de red (protocolo encapsulado) y el protocolo de nivel de enlace.



Figura 8: Situación de MPLS en el modelo de referencia OSI

La cabecera de un paquete MPLS se encuentra también entre estos dos niveles, es decir, entre la cabecera del nivel de enlace y la cabecera del nivel de red. De esta forma, para el protocolo de nivel de enlace, un paquete MPLS serán datos empaquetados del nivel superior del modelo.

3.1. Terminología

Para poder explicar el funcionamiento de MPLS es necesario definir varios términos propuestos por el IETF para ello:

LER *Layer Edge Router*. Router frontera entre capas. Es el encaminador que se encuentra en el borde del área MPLS y el encargado de añadir las cabeceras MPLS entre las cabeceras de red y de enlace del paquete entrante. Además, es el encargado de retirar dicha información cuando un paquete sale de la zona MPLS.

LSR *Label Switch Router*. Conmutador de etiquetas. Se trata del conmutador del interior de la zona MPLS que interpreta el valor de la cabecera MPLS y la modifica si es necesario, pero no añade ni elimina las cabeceras MPLS.

FEC *Forward Equivalence Class*. Clase de envío equivalente vista anteriormente. Es el conjunto de paquetes o flujos de información que ingresan por un mismo LER y a los cuales éste les asigna la misma etiqueta.

LSP *Label Switched Path*. Camino conmutado de etiquetas. Se trata del camino que describen el conjunto de encaminadores y conmutadores que atraviesan los paquetes de un FEC concreto en un único nivel jerárquico en lo que a la zona MPLS se refiere.

Label Etiqueta. Información o cabecera que se añade a un paquete cuando ingresa en la zona MPLS y que se elimina cuando sale de ésta.

LS *Label Stack*. Es la pila de etiquetas, es decir, un conjunto de etiquetas dispuestas en forma de pila utilizada para permitir la característica de controlar la existencia de zonas MPLS dentro de otras zonas MPLS.

MPLS Domain Dominio MPLS. Se trata del conjunto de encaminadores contiguos capaces de trabajar con MPLS para enrutamiento y conmutado, y que se encuentran dentro de un mismo ámbito administrativo.

Label Merging Fusión de etiquetas. Reemplazo de múltiples etiquetas de entrada para una FEC particular por una sola etiqueta de salida.

Label Switched Hop Salto de conmutación de etiquetas. Salto entre dos nodos MPLS en los que el reenvío se hace usando etiquetas.

Merge Point Punto de fusión. Nodo en el que se realiza la fusión de etiquetas.

VC Merge Fusión de circuitos virtuales. Fusión de etiquetas en donde la etiqueta MPLS se transporta en el campo ATM VPI/VCI. De esta forma, se permite que múltiples circuitos virtuales se fusionen en un único circuito virtual.

VP Merge Fusión de caminos virtuales. Fusión de etiquetas en donde la etiqueta MPLS se transporta en el campo ATM VPI. De este modo, se permite que múltiples caminos virtuales se fusionen en uno sólo. Dos células con el mismo valor VCI se han originado en el mismo nodo.

En MPLS, la asignación de un paquete a una FEC se realiza cuando el paquete entra en la red asignándole a dicho paquete una etiqueta. En los siguientes saltos sólo se usará la etiqueta para determinar la interfaz por donde reenviar el paquete, por lo que no será necesario analizar la cabecera del nivel de red. La etiqueta se usa como índice en la tabla de encaminamiento donde se obtiene el siguiente salto y la nueva etiqueta con la que sustituir la anterior. Hay que recordar que las etiquetas son locales a los encaminadores. En MPLS, los conmutadores pueden realizar el reenvío, pero éstos no tienen necesidad de analizar las cabeceras del nivel de red. Las ventajas de basar el reenvío en las etiquetas en vez de en la cabecera del nivel de red se muestran a continuación:

- Dado que un paquete se asigna a una FEC cuando entra en la red, el encaminador frontera que encapsula el paquete podrá usar toda la información que tenga sobre el paquete, incluso información que no esté en la cabecera del nivel de red. Por ejemplo, podrá usar información del nivel de transporte, como los números de puerto, para asignar paquetes a los FEC. Por tanto, gran parte del trabajo se realiza antes de que el tráfico entre en la red.
- Un paquete que entra en la red por un determinado encaminador puede etiquetarse de distinta forma que si hubiera entrado por otro, de forma que se pueden tomar decisiones dependientes del encaminador frontera que encapsula el paquete. Esto no se puede hacer en el encaminamiento convencional porque la identidad del encaminador frontera que introdujo el paquete en la red no viaja con el paquete.
- Se podría forzar a un paquete a seguir una ruta elegida explícitamente antes o en el momento que el paquete entre en la red, en vez de elegirse por el algoritmo dinámico de encaminamiento a medida que el paquete fluye por la red. Esto podría hacerse para permitir la ingeniería de tráfico (MPLS-TE). En el encaminamiento convencional, el paquete tendría que llevar la información de la ruta (encaminamiento fuente), mientras que en MPLS se puede usar una etiqueta para representar la ruta, de tal forma que el paquete no tiene por qué llevar la información de la misma.

Algunos encaminadores analizan la cabecera del nivel de red para determinar la clase de servicio a la que pertenece el paquete así como para determinar el siguiente salto. Con la información de la clase de servicio, el encaminador podrá o no aplicar alguna disciplina planificada a los paquetes. MPLS permite, pero no impone, que la clase de servicio se infiera total o parcialmente de la etiqueta; así podremos decir que una etiqueta representa la combinación de una FEC y una clase de servicio.

Como vimos en el apartado de conceptos básicos de la conmutación de etiquetas, MPLS se llama así porque soporta cualquier protocolo de nivel de red así como cualquiera de nivel de enlace. Por tanto, MPLS puede o no usar tecnologías subyacentes de backbone como ATM, Frame Relay, SDH y DWDM.

3.2. Tipos de nodos MPLS

Los LSR frontera son los encargados de etiquetar los paquetes que entran en la red. Para poder realizar este trabajo, estos LSR deben implementar el componente de control y el componente de reenvío tanto del encaminamiento convencional como de la conmutación de etiquetas.

Si un paquete entra en la red, el encaminador frontera utilizará el componente de reenvío de la conmutación de etiquetas para determinar la etiqueta que ponerle al paquete. Si el siguiente salto no es un LSR y el paquete no tiene etiqueta, entonces el LSR deberá reenviar el paquete usando el componente de reenvío del encaminamiento convencional.

Cuando el paquete va a salir de la red MPLS, el LSR que recibe el paquete le quitará la etiqueta y lo reenviará al siguiente salto usando el componente de reenvío del encaminamiento convencional. Dicho LSR sabrá que el paquete quiere abandonar la red simplemente porque el siguiente salto no es un LSR.

Los tipos de nodos MPLS son los siguientes [7]:

- LSR de entrada (*ingress LSR*). LSR que recibe tráfico de usuario (por ejemplo datagramas IP) y lo clasifica en su correspondiente FEC. Genera una cabecera MPLS asignándole una etiqueta y encapsula el paquete junto a dicha cabecera obteniendo una PDU MPLS (*Protocol Data Unit, Unidad de Datos de Protocolo*).
- LSR de salida (*egress LSR*). LSR que realiza la operación inversa al de entrada, es decir, desencapsula el paquete eliminando la cabecera MPLS.
- LSR intermedio o interior. LSR que realiza el intercambio de etiquetas examinando exclusivamente la cabecera MPLS (obtiene la etiqueta para poder realizar la búsqueda en la tabla de encaminamiento).

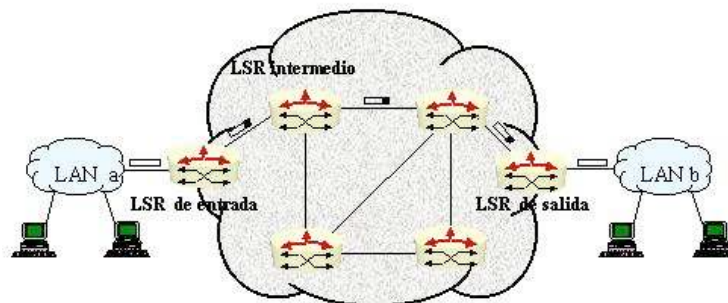


Figura 9: Tipos de nodos MPLS

3.3. Protocolos de distribución de etiquetas

Un protocolo de distribución de etiquetas es un conjunto de procedimientos por los que un LSR le informa a otro de las asociaciones de etiquetas que ha hecho a las diferentes FEC.

A dos LSR que utilizan un protocolo de distribución de etiquetas para intercambiar información de asociaciones de etiquetas a las FEC se les conoce como un par de distribución de etiquetas (*label distribution peers*) respecto a la información de las asociaciones que intercambian.

MPLS no asume que haya sólo un protocolo de distribución de etiquetas. De hecho, se están normalizando distintos protocolos de distribución de etiquetas, como por ejemplo LDP (*Label Distribution Protocol*).

3.3.1. Distribución y asignación de etiquetas

En MPLS, la decisión correspondiente a la asignación de una etiqueta a una FEC la realiza el LSR que está río abajo (*downstream*) con respecto a la asociación.

El LSR que está río abajo informa al LSR que está río arriba de la asociación. Por tanto, las etiquetas se asignan o asocian río abajo y se distribuyen en el sentido que va del LSR que está río abajo al LSR que está río arriba. MPLS permite variaciones en la asociación río abajo:

Río abajo solicitado (downstream-on-demand)

Un LSR le solicita explícitamente a su siguiente salto una asociación de una etiqueta a una FEC.



Figura 10: Río abajo solicitado

Río abajo no solicitado (unsolicited-downstream)

Un LSR distribuye asociaciones a otros LSR que no lo han solicitado explícitamente.

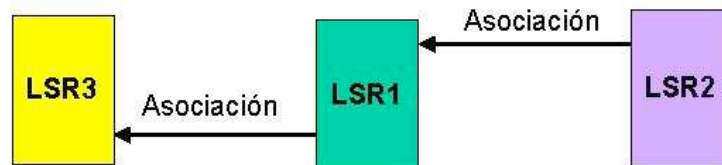


Figura 11: Río abajo no solicitado

Estas aproximaciones se pueden usar por separado o conjuntamente. En caso de usarlas conjuntamente en una adyacencia de distribución de etiquetas (cuando tenemos dos LSR que son pares de distribución de etiquetas), ambos LSR se tendrán que poner de acuerdo en la técnica a usar.

3.3.2. Formato de las etiquetas

Una etiqueta MPLS tiene 32 bits y, como hemos visto anteriormente, se sitúa entre la cabecera de nivel 2 y la de nivel 3 [9]. El formato es el siguiente, donde todos los campos son de tamaño fijo:



Figura 12: Formato de las etiquetas

- Etiqueta Es el valor de la etiqueta MPLS.
- Exp Estos bits están reservados para uso experimental. Algunos artículos sobre servicios diferenciados (DiffServ) discuten su uso.
- S Bit de apilamiento (*Stacking bit*). Se usa para apilar etiquetas del siguiente modo: cuando está a 1 indica que esta cabecera MPLS es la última que hay antes de encontrarse con la redundancia de red. Si por el contrario se encuentra a 0, indica que tras esta cabecera MPLS se encuentra otra cabecera MPLS y no la cabecera de red.

TTL Tiempo de vida (*Time To Live*). Número de nodos (saltos) que puede atravesar el paquete MPLS. Es necesario porque los LSR intermedios no analizan el campo IP TTL.

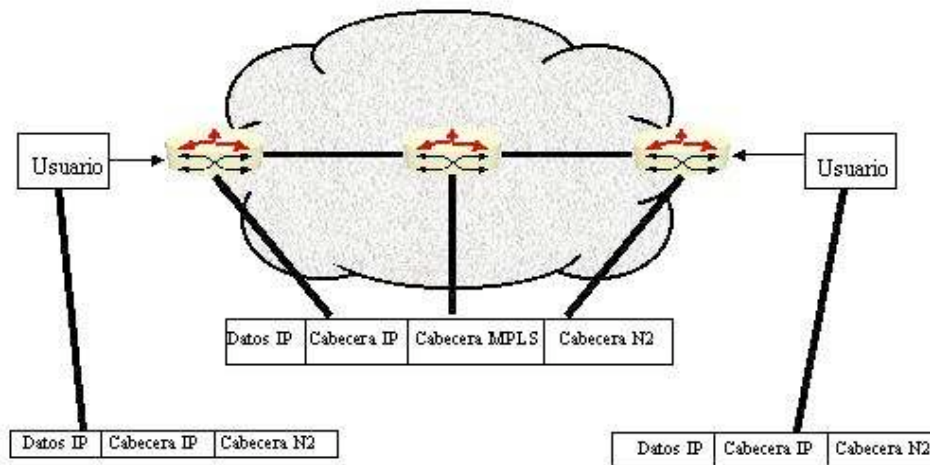


Figura 13: Esquema MPLS con los distintos paquetes que entran en juego

3.3.3. La pila de etiquetas

Como hemos visto, un paquete en MPLS puede tener más de una etiqueta, organizadas a modo de pila (FIFO), que es a lo que se conoce como pila de etiquetas (LS).

Aunque MPLS soporte una jerarquía gracias a la pila de etiquetas, el procesamiento de un paquete etiquetado es completamente independiente del nivel de la jerarquía. Siempre que se procese una etiqueta, ésta será la de la cima, sin importar cuántas etiquetas pueda haber debajo. Por otro lado, se puede considerar a un paquete no etiquetado como un paquete con una pila de etiquetas vacía.

Si la profundidad de la pila de etiquetas de un paquete es m , a la etiqueta que está al fondo de la pila se le llama etiqueta de nivel 1, a la que está encima etiqueta de nivel 2, y así sucesivamente [7].

Veamos un ejemplo: En la siguiente figura tenemos tres dominios. Supongamos que el dominio 2 es un dominio de tránsito y que en dicho dominio no se originan paquetes y que tampoco hay paquetes destinados a él. Para anunciar las direcciones del dominio 3, el LSR F le distribuye la información al LSR E, éste distribuye la información al LSR B el cual se la distribuye al LSR A. Por último, decir que no se distribuye la información a los LSR C y D porque son LSR interiores.

En el siguiente ejemplo se usan dos niveles de etiquetas. Cuando el tráfico entra en el segundo dominio se apila una nueva etiqueta en la cima de la pila, por lo que las etiquetas que hubiera en la pila descienden un nivel.

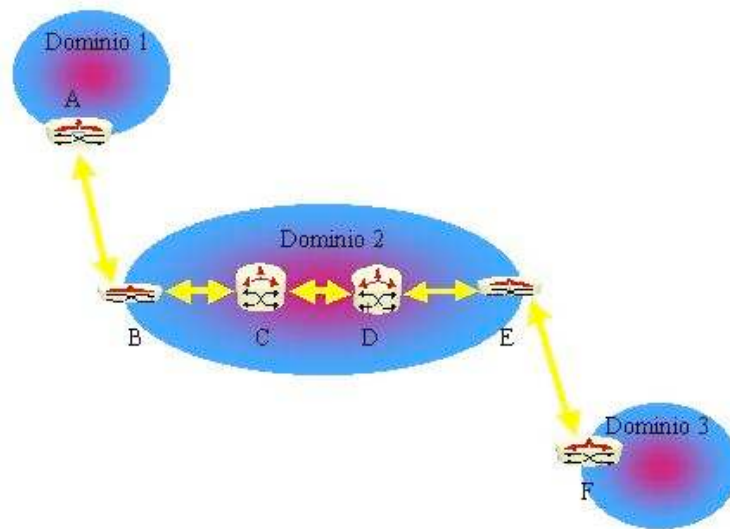


Figura 14: Ejemplo de pila de etiquetas

3.4. Aplicaciones de MPLS

Las aplicaciones principales de MPLS son las siguientes [9]:

- Encaminamiento explícito e ingeniería de tráfico.
 - Soporte a las CoS.
- Servicio de redes privadas virtuales (VPN, *Virtual Private Network*).
- Integración de IP con todo tipo de redes subyacentes: Frame Relay, ATM, SDH y DWDM.

3.4.1. Ingeniería de tráfico

La ingeniería de tráfico persigue adaptar flujos de tráfico a recursos físicos de la red, de tal forma que exista un equilibrio entre dichos recursos. Con ello se conseguirá que no haya recursos excesivamente utilizados, con cuellos de botella, mientras existan recursos poco utilizados.

Uno de los mayores problemas de las redes IP actuales es la dificultad de ajustar el tráfico IP para hacer un mejor uso del ancho de banda, así como mandar flujos específicos por caminos específicos. En las redes IP convencionales, los paquetes suelen seguir el camino más corto, política que siguen los protocolos de encaminamiento interior. Esto suele provocar que algunos enlaces se saturen mientras otros están infrautilizados. Dicho problema se ha venido resolviendo añadiendo más capacidad a los enlaces. Veamos un ejemplo:

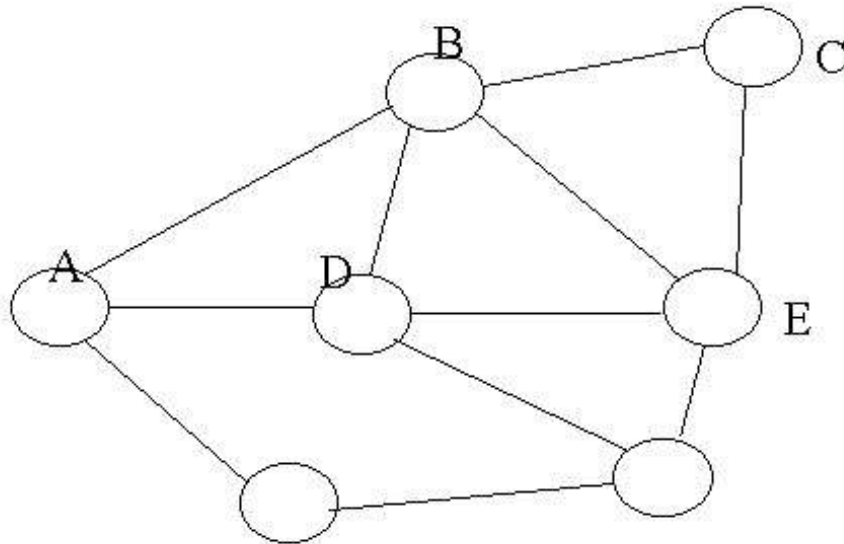


Figura 15: Topología de red para explicar la ingeniería de tráfico

El camino más corto entre A y C según la métrica normal IGP es el que tiene dos saltos (A-B-C), pero puede que el exceso de tráfico sobre estos enlaces o la carga de los encaminadores hagan aconsejable la utilización de un camino que requiera saltos adicionales, como por ejemplo A-D-E-C.

MPLS es una herramienta efectiva para la ingeniería de tráfico, ya que:

- Permite al administrador de la red establecer rutas explícitas, especificando el camino físico exacto de un LSP.
- Permite obtener estadísticas de uso de un LSP.
- Permite usar el encaminamiento basado en restricciones de modo que el administrador de la red pueda seleccionar determinadas rutas para servicios especiales con distintos niveles de calidad (con garantías de ancho de banda, etc.).

3.4.2. Soporte a las clases de servicio (CoS)

MPLS soporta diferentes clases de servicio para cada LSP. Como caso particular, puede soportar servicios diferenciados (DiffServ) en el mismo LSP.

Históricamente, Internet ha ofrecido un solo nivel de servicio denominado *Best Effort*. Con la aparición de aplicaciones multimedia y aplicaciones en tiempo real, surgió la necesidad de la diferenciación de servicios en Internet. De esta forma, se pueden diferenciar servicios como el correo electrónico de otros que dependen mucho más del retardo y de la variación del mismo como el video y la voz interactiva.

El modelo de los servicios diferenciados define los mecanismos para poder clasificar el tráfico en clases de servicio con diferentes prioridades. Para clasificar el tráfico se emplea el campo ToS (*Type of Service*, Tipo de Servicio), el cual es denominado DS en DiffServ. Una vez clasificados los paquetes en la frontera de la red, los paquetes se reenvían basándose en el campo DS; dicho reenvío se realiza por salto, es decir, el nodo decide por sí solo cómo se deberá realizar el reenvío. A este concepto se le denomina comportamiento por salto (PHB, *Per-Hop Behavior*).

MPLS se adapta bien a este modelo, ya que las etiquetas MPLS tienen el campo *Exp* para poder propagar la clase de servicio CoS en el correspondiente LSP. Por tanto, una red MPLS puede transportar distintas clases de tráfico y entre cada par de LSR exteriores se pueden tener distintos LSP con distintas prestaciones y distintos anchos de banda.

3.5. MPLS-TE

Como se ha comentado anteriormente, MPLS ha sido dotado de extensiones de ingeniería de tráfico, con lo cual, en la actualidad, se cuenta con MPLS-TE (*Multiprotocol Label Switching Traffic Engineering*), que proporciona capacidad para equilibrar la carga en la red, tolerancia a fallos, reencaminamiento rápido, establecimiento y cálculo de LSP de respaldo, recuperación de LSP fallidos y de paquetes descartados, etc.

3.6. GMPLS

El esquema de MPLS tradicional permite algo realmente innovador: integrar en una única red de transporte tecnologías tan dispares como IP, ATM y Frame Relay, lo que permite ahorrar el coste de inversión y actualización de las redes actuales a la vez que provee de los mecanismos necesarios para poder realizar ingeniería de tráfico, especificar clases de servicios, etc.

Sin embargo, la tecnología MPLS no permite tener control ni aplicar técnicas de ingeniería de tráfico sobre redes de conmutación por división de longitud de onda (DWDM), de multiplexación por división de tiempo (TDM), etc., puesto que MPLS se encuentra por encima del nivel en que trabajan este tipo de redes y no tiene métodos de señalización ni encaminamiento para ellas. Ha sido necesario rediseñar MPLS y un gran número de protocolos de todos los niveles implicados para ser capaz de adaptarse a este tipo de redes con tecnologías tan distintas y poder situar a MPLS un nivel más abajo, incluso en el nivel físico. A esta tecnología rediseñada a partir de MPLS y de MPλS (un primer acercamiento de MPLS a las redes ópticas) es a la que se llama *Generalized Multiprotocol Label Switching* (GMPLS) o conmutación de etiquetas multiprotocolo generalizado.

GMPLS surge de forma natural como una evolución de MPLS y de MPλS. No olvidemos que MPLS se encuentra en el nivel 2+ de la pila de protocolos. Sin embargo, poco a poco las redes tienden a simplificarse a medida que aumenta la calidad de las mismas compactando los niveles definidos por el modelo de referencia OSI de la ISO hasta límites insospechados.

Aún así, el surgir de las redes ópticas ha cambiado mucho el mundo de los protocolos de comunicaciones. Las redes ópticas no están pensadas, diseñadas, ni preparadas para poder inspeccionar el contenido de la información que transportan por lo que difícilmente se podrá conmutar en base a dicha información. Más bien se hacen necesarios dispositivos de nivel físico capaces de conmutar en base a la señal portadora de la información.

Surgen aquí un sinnúmero de problemas, como el hecho de la creación de etiquetas para los paquetes GMPLS, el soporte de distintos tipos de conmutación a nivel físico, de enlace o de red, problemas de compatibilidad entre dispositivos de estos tipos, etc.

La primera de las cuestiones era definir por tanto los tipos de interfaces a través de los cuales GMPLS podría conmutar la información, cada uno de estos tipos asociado a un nivel concreto y a una tecnología, pero todo ello regulado para poder estandarizar GMPLS con éxito. De este modo, se podrá construir un superdominio GMPLS que contenga subredes de distinta tecnología y el tráfico pueda fluir de extremo a extremo de forma transparente usando GMPLS como tecnología integradora en todo su conjunto.

Los interfaces comentados son los siguientes:

- *Packet Switching Capable* (PSC)
- *TDM Switching Capable* (TSC)
- *Lambda Switching Capable* (LSC)
- *Fiber Switching Capable* (FSC)

El funcionamiento general de GMPLS es similar al de MPLS, puesto que es una generalización de éste. Como se ha definido para los interfaces anteriores, entre dispositivos PSC puede haber un LSP, al igual que entre dispositivos TSC, LSC y FSC. Esto supone que GMPLS actúa en todos los niveles existentes de la comunicación. Pero más allá, GMPLS permite también crear LSP entre nodos heterogéneos (y por tanto redes de tecnologías distintas), y es aquí donde se muestra la mayor funcionalidad de GMPLS.

Por tanto, GMPLS permite el establecimiento de un dominio MPLS con distintos dispositivos de diferentes tecnologías trabajando a niveles distintos, de forma simultánea y transparente, siempre y cuando los dispositivos cuenten con interfaces englobadas dentro de alguno de los cuatro tipos anteriores. La única restricción es que las interfaces origen y destino de los LSP creados tienen que tener la misma capacidad, es decir, ser del mismo tipo (PSC, TSC, LSC o FSC).

4. BGP

Ya sabemos que el tráfico circulante por una red usa para guiarse protocolos de encaminamiento interior como IS-IS u OSPF. Sin embargo, los protocolos de encaminamiento interior no saben cómo guiar el tráfico más allá del límite de la red sobre la que actúan, es decir, mientras en el encaminamiento interior la entidad organizativa tiene el control completo de una comunicación desde el origen al destino, no ocurre lo mismo en el encaminamiento exterior o interdominio, donde las comunicaciones salen fuera del sistema autónomo (AS).

Los AS se necesitan los unos a los otros, ya que una comunicación puede extenderse sobre varios de ellos. Sin embargo, las entidades organizativas no desean que se conozca cómo se llevan a cabo las acciones que realizan dentro de su AS y cada uno de ellos solamente tiene una visión parcial de la topología de Internet, la de su propia red.

4.1. Evolución histórica de BGP

Hace décadas, los administradores de los AS se enviaban periódicamente un fichero de rutas para configurar de forma manual las rutas entre los distintos AS. Se trataba de tiempos en los que Internet estaba formada por centros de investigación y era un entorno de confianza.

Posteriormente, y ante el crecimiento de la red de redes, apareció GGP (Gateway to Gateway Protocol), con lo que en cada AS existían servidores de ruta que intercambiaban las rutas entre ellos de forma periódica.

En el año 1984, el IETF creó EGP (Exterior Gateway Protocol) [10] [11], cuyas características eran las siguientes:

- Cada AS solamente puede anunciar rutas que lleven sus propias redes.
- Tiene poca capacidad para aplicar políticas.
- Tiene facilidad para crear bucles.
- Entiende Internet como una red muy jerárquica.

En el año 1989, el IETF creó el RFC de la primera versión de BGP [12], que comparte muchas de las ideas de EGP pero corrige ciertos problemas encontrados, puesto que permite eliminar bucles, que un AS pueda anunciar rutas propias así como rutas aprendidas de terceros AS o políticas de encaminamiento más avanzadas.

Por último, y desde 1990 hasta la actualidad, el IETF ha modificado BGP y ha liberado las versiones 2, 3 y 4, añadiendo al estándar del protocolo mejoras a problemas que han ido surgiendo con la expansión de Internet, como son las mejoras añadidas en cuanto a la escalabilidad y al tiempo de convergencia u otras características que lo acercan a la Internet real, menos jerarquizada de lo que se suponía en un principio.

4.2. Funcionamiento de BGP-4

BGP es un protocolo que mantiene tablas de rutas completas para llegar a un destino concreto (*path-vector*) y la información de estas tablas se retransmite a los nodos con los que comparte información mediante un conjunto de mensajes BGP [13].

Cada una de las rutas incorpora una serie de datos:

- Red destino
- Conjunto de AS por los que hay que pasar para llegar a esa red
- Atributos de la ruta
- Otros datos menores

Las rutas son significativamente distintas a la información transmitida por los protocolos de encaminamiento interior. De esta forma, el modo de funcionamiento y los problemas asociados son también diferentes.

Los mensajes que utiliza el protocolo se envían utilizando conexiones TCP. Los distintos tipos de mensajes que maneja este protocolo son los siguientes:

Apertura (*Open*). Se utiliza para establecer una relación de vecindad con otro encaminador y así compartir información de encaminamiento.

Actualización (*Update*). Se utiliza para transmitir información a través de una ruta y/o enumerar múltiples rutas que se van a eliminar.

Mantenimiento (*Keepalive*). Utilizado para confirmar un mensaje Open y para confirmar periódicamente la relación de vecindad. En otras palabras, sirve para mantener con vida una sesión BGP.

Notificación (*Notification*). Este tipo de mensajes se envía cuando se detecta una condición de error.

Vemos que no existe un mensaje para cerrar la sesión BGP (*close*), ya que cuando un nodo BGP cierra la sesión, se produce un mecanismo de actualización en cascada que puede afectar a todo Internet. Durante este proceso, la red es incoherente y el tráfico circulante se puede perder; por ello, una sesión BGP únicamente se cierra cuando hay un error irreparable o cuando un nodo BGP se va a desactivar para siempre.

Existen varios pasos que el protocolo BGP lleva a cabo. Estos son los siguientes:

1. *Adquisición de vecinos.* Ocurre cuando dos encaminadores situados en diferentes AS se ponen de acuerdo para intercambiar información de encaminamiento regularmente. Un encaminador le enviará a otro un mensaje Open. Si el destino acepta la solicitud le devolverá un mensaje de mantenimiento (Keepalive).
2. *Detección de vecino alcanzable.* Una vez realizada la adquisición de vecinos se utiliza este procedimiento para mantener la relación. Periódicamente ambos dispositivos de encaminamiento se envían mensajes de mantenimiento para asegurarse que su par sigue existiendo y que desea continuar con la relación de vecindad.
3. *Detección de red alcanzable.* Cada encaminador mantiene una base de datos con las redes que puede alcanzar y la ruta preferida para alcanzar dichas redes. Cuando se realiza un cambio en esta base de datos, el encaminador enviará un mensaje de actualización por difusión. De esta forma el resto de los encaminadores BGP podrán construir y mantener la información de encaminamiento.

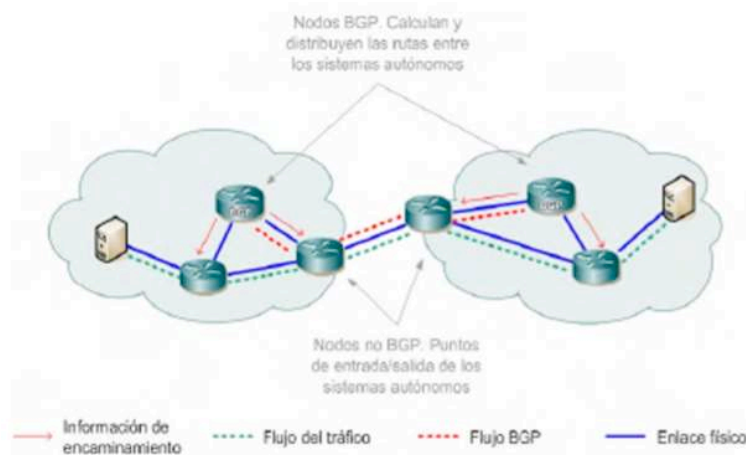


Figura 16: Esquema general del funcionamiento de BGP

A continuación se explica de forma resumida el funcionamiento de BGP:

Ya hemos dicho que un nodo BGP tiene como misión intercambiar con otros nodos BGP información de accesibilidad a nivel de red que permita conocer la trayectoria que debe seguir un tráfico concreto de red para llegar a un destino. Puede haber más de un nodo BGP en cada AS; además, no necesariamente los nodos BGP tienen por qué ser los encaminadores IP de entrada al sistema autónomo.

El nodo BGP debe difundir las rutas que aprenda de sus vecinos (en otros AS o en el suyo propio) a los encaminadores del interior del sistema autónomo local, puesto que para ello precisamente está diseñado BGP. De esta forma existen mecanismos para que, de alguna manera, BGP interactúe con los protocolos de encaminamiento intradominio, por ejemplo con OSPF [14].

Dado que BGP es el estándar por excelencia en el encaminamiento interdominio en Internet, hoy por hoy casi todos los sistemas autónomos incorporan nodos BGP. Sólo algunos sistemas autónomos poco actualizados siguen utilizando EGP o algunas versiones antiguas de BGP.

BGP necesita un protocolo de nivel de transporte que sea fiable. En su diseño, se delegó esta función al protocolo TCP, por lo que siempre es necesario que una sesión BGP funcione sobre una conexión TCP previamente establecida. De este modo, no es necesario volver a implementar mecanismos de fiabilidad, reordenación, segmentación, secuencia, etc.

Un nodo BGP conoce la existencia de un vecino BGP y su dirección mediante la configuración manual del administrador de la red. Esto es así por las políticas seguidas en cuanto al intercambio de información de accesibilidad y de transporte del tráfico. Dos organizaciones acuerdan mediante un contrato escrito el intercambio de este tipo de información y se facilitan toda la información necesaria para configurar sus respectivos nodos BGP. A partir de este momento, los nodos se reconocen como vecinos y establecen entre ellos una conexión TCP. Sobre ésta, se crea una sesión BGP mediante el intercambio de mensajes Open (y su correspondiente mensaje Keepalive de respuesta), momento tras el cual se sondean ambos nodos para detectar las características que soportan y llegar a un acuerdo de mínimos.

Una vez establecida la sesión BGP, el primer paso que dan los nodos es intercambiarse la tabla completa de encaminamiento. Una tabla de encaminamiento interdominio puede superar fácilmente en la actualidad las 120.000 entradas, las cuales contienen una ruta completa compuesta por sus atributos. Por otro lado, cada ruta BGP está compuesta no sólo por el siguiente salto IP que hay que dar para llegar hacia un destino, sino por el identificador de todos y cada uno de los sistemas autónomos que componen la ruta. Puesto que cada AS puede contener miles de encaminadores, las rutas solamente están compuestas por los identificadores de AS (y no por todos los encaminadores IP que hay que pasar) y los saltos en la ruta son, por tanto, de una granularidad superior. En cualquier caso, debido a que únicamente es posible anunciar una ruta válida en cada mensaje Update, es obvio que harán falta decenas de miles de mensajes Update para poder intercambiar las tablas de encaminamiento.

Una vez que los vecinos BGP se hayan intercambiado las tablas de encaminamiento y este cambio haya sido correctamente propagado a otros sistemas autónomos, la red se estabiliza, el tráfico puede circular por las rutas calculadas y todo funciona correctamente. Se dice entonces que la red ha convergido. Los cambios en las tablas de encaminamiento de un AS no solo afectan a los dos AS implicados en el intercambio, puesto que un nodo BGP puede hacer llegar a un tercer AS información de encaminamiento aprendida de otras sesiones BGP, y éstos pueden hacer lo mismo a su vez. Así que es importante entender que un cambio en las tablas de encaminamiento puede afectar a todo Internet.

A partir del momento en que los vecinos BGP se intercambian las tablas completas de encaminamiento, cualquier necesidad de modificar una ruta concreta, añadir nuevas o eliminar rutas previamente anunciadas como válidas, se realiza de la misma forma: mediante mensajes Update. La diferencia estriba en que, en este caso, con un simple mensaje Update se realizará la actualización (incremental) de las tablas de encaminamiento interdominio.

En principio, se supone que no se debe cerrar la sesión BGP puesto que el mantenimiento de las tablas de encaminamiento interdominio es vital para el funcionamiento de grandes áreas de la red y perdura en el tiempo. Sin embargo, ante cualquier fallo, se envía un mensaje de tipo Notification y se procede al cierre de la sesión BGP, se eliminan todas las rutas asociadas a esa sesión BGP y se cierra la conexión TCP sobre la que se creó dicha sesión. Si se reinicia una sesión BGP, se tienen que volver a intercambiar las tablas completas de encaminamiento hasta que la red converja.

4.3. MPLS-BGP

En MPLS se puede utilizar BGP para distribuir la información de asociación de etiquetas para cada ruta que se anuncie. Esto es posible gracias a las extensiones multiprotocolo (*MPEs: Multiprotocol Extensions*) de BGP versión 4 [15].

Para distribuir las etiquetas se utilizan los mensajes de actualización (utilizando *piggybacking*), los cuales también se utilizan para distribuir la información de las rutas. La etiqueta se codifica en el campo NLRI (*Network Layer Reachability Information*, información de alcanzabilidad del nivel de red) y para indicar que el campo NLRI contiene una etiqueta, se utiliza el campo SAFI (*Subsequent Address Family Identifier*, identificador de familias de direcciones consecutivas). Un hablante BGP no podrá utilizar BGP para la distribución de etiquetas hacia un igual a no ser que dicho igual le indique que puede procesar mensajes de actualización con el campo SAFI especificado.

Ventajas de la utilización de MPLS-BGP:

- Si dos LSR adyacentes también son hermanos BGP (*peers*), entonces la distribución de etiquetas se puede realizar sin necesidad de tener otro protocolo de distribución de etiquetas.
- Supongamos una red con dos clases de LSR: LSR exteriores, que hacen de interfaz con otras redes, y LSR interiores, los cuales sólo transmiten tráfico entre los LSR exteriores. Si los LSR exteriores también son hablantes BGP y distribuyen etiquetas MPLS con la información de encaminamiento, entonces los LSR interiores no necesitan recibir ninguna de las rutas BGP de los hablantes BGP.

Como se comentó anteriormente, las etiquetas se transportan como parte del campo NLRI en los atributos de extensión multiprotocolo. El AFI indica la familia de direcciones de la ruta asociada. Si el campo NLRI contiene una etiqueta, se le dará un valor de cuatro al campo SAFI para identificar esta situación.

El campo NLRI se codifica en una o más tripletas <longitud, etiqueta, prefijo>de la siguiente forma:

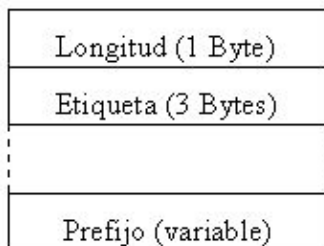


Figura 17: Campo NLRI

- | | |
|----------|--|
| Longitud | Este campo se utiliza para indicar la longitud en bits del prefijo de dirección más la etiqueta. |
| Etiqueta | El campo de la etiqueta sirve para transportar una o más etiquetas (lo que corresponde a la pila de etiquetas). Cada etiqueta se codifica en 3 Bytes, donde los 20 bits de más peso contienen el valor de la etiqueta y los bits de menos peso contienen la parte baja de la pila. |
| Prefijo | Este campo contiene los prefijos de dirección seguidos de bits de relleno para conseguir que el campo ocupe un número exacto de Bytes. |

Para retirar una ruta anunciada previamente un hablante BGP podrá:

- Anunciar una nueva ruta (y una etiqueta) con la misma NLRI que la ruta previa.
- Listando la NLRI de la ruta previa en el campo de retirada de rutas (*Withdrawn Routes Field*) de un mensaje de actualización (Update).

Si se termina una sesión BGP también se retiran todas las rutas anunciadas previamente.

4.4. Situaciones de BGP

A continuación se muestran varias situaciones que se llevan a cabo en BGP.

4.4.1. Anuncio de múltiples rutas a un destino

Un hablante BGP puede mantener (y anunciar a sus hermanos) más de una ruta hacia un mismo destino siempre que cada ruta tenga sus propias etiquetas.

La codificación mencionada previamente permite que un solo mensaje de actualización contenga múltiples rutas, cada una con sus propias etiquetas.

Para el caso en el que un hablante BGP anuncie múltiples rutas a un destino, si la ruta es retirada y la etiqueta se especifica a la vez que la retirada, sólo dicha ruta con su correspondiente etiqueta es retirada. Si la ruta se retira y no se especifica etiqueta, entonces sólo la ruta sin etiquetar correspondiente se retira y se mantienen las rutas etiquetadas.

4.4.2. Hermanos BGP que no son adyacentes

A continuación se muestra un ejemplo:



Figura 18: Ejemplo de distribución de etiquetas

D le distribuye a A la etiqueta L. A no podrá simplemente apilar L en la pila de etiquetas del paquete y enviar dicho paquete hacia B. D debe ser el único LSR que vea L en la cima de la pila. Antes de que A le envíe el paquete deberá apilar otra etiqueta que habrá obtenido previamente de B. B reemplazará esta etiqueta con otra que obtuvo de C. Dicho de otra forma, de haber un LSP entre A y D. Si no existiera dicho LSP, A no podría usar la etiqueta L. Esto siempre será cierto cuando las etiquetas se distribuyan entre LSR que no son adyacentes, no importando si la distribución se hace por BGP o por cualquier otro método.

4.5. Problemática de BGP

BGP presenta varios problemas a resolver en la actualidad, que se pueden clasificar en tres tipos [16]:

Problemas de convergencia. El tiempo de convergencia es aquél que la red tarda en ser coherente tras un cambio. El cambio producido por un mensaje Update se propaga por toda la red. Mientras, la red es incoherente. El tiempo de convergencia debe tender a cero.

Problemas de escalabilidad. BGP tiene diversas restricciones que lo hacen escalar no demasiado bien. La mayor parte de ellas vienen dadas por el hecho de que un AS tenga más de una conexión con otros AS vecinos.

Problemas de ingeniería de tráfico. Para la ingeniería de tráfico es importante tener el control de todos los lugares por los que va a circular el tráfico. En BGP no es sencillo, y no existe BGP-TE como tal.

En cualquier caso, todos los factores están interrelacionados, por lo que es complejo definir los límites en que se ve afectado BGP.

5. RSVP

El protocolo de reserva de recursos (*RSVP*, *Resource reSerVation Protocol*) se utiliza para reservar recursos para una sesión en un entorno de red IP. Se trata de un protocolo de estado blando como veremos posteriormente. En el siguiente gráfico podemos apreciar que el protocolo RSVP se apoya en IP:

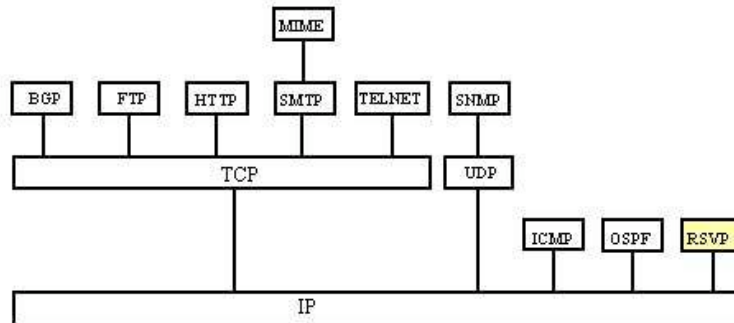


Figura 19: Situación de RSVP en la familia de protocolos TCP/IP

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.

5.1. Características de RSVP

En los siguientes puntos se muestran las características de RSVP:

- Se diseña para trabajar con cualquier servicio de QoS (los objetos propios de la QoS no están definidos por el protocolo).
- Permite Unicast y Multicast. No es un protocolo de encaminamiento, sino que está pensado para trabajar conjuntamente con éstos.
- No transporta datos de usuario.
- Los protocolos de encaminamiento determinan dónde se reenvían los paquetes mientras que RSVP se preocupa por la QoS de los paquetes reenviados de acuerdo con el encaminamiento.
- Es un protocolo simplex: petición de recursos sólo en una dirección, diferencia entre emisor y receptor. El intercambio entre dos sistemas finales requiere de reservas diferenciadas en ambas direcciones.
- Reserva iniciada por el receptor (protocolo orientado al receptor).
- Mantenimiento del estado de la reserva (estado blando) en los encaminadores. El mantenimiento de la reserva es responsabilidad de los usuarios finales.
- Permite diferentes tipos de reservas.
- Protocolo transparente para los encaminadores no RSVP.
- Soporta IPv4 e IPv6 aunque no sea un protocolo de transporte.

Existen dos tipos fundamentales de mensajes RSVP:

Mensajes Path Estos mensajes los generan los emisores y los utilizan para establecer el camino de la sesión. Describen el flujo del emisor y proporcionan la información del camino de retorno hacia el mismo. Este tipo de mensajes pueden atravesar encaminadores que no entiendan RSVP puesto que tienen una dirección IP origen y una dirección IP destino.

Mensaje Resv Estos mensajes los generan los receptores y sirven para hacer una petición de reserva de recursos. Crean el "estado de la reserva" en los encaminadores. Generalmente, una petición de recursos implicará una reserva de éstos en todos los nodos del camino del flujo de datos. Estos mensajes siguen exactamente el camino inverso al de los datos.

Por tanto, el mensaje Path es el responsable del inicio de la operación y es mandado a los participantes potenciales de la sesión. El mensaje Resv se manda en respuesta al mensaje Path. La siguiente figura muestra el uso de estos mensajes:

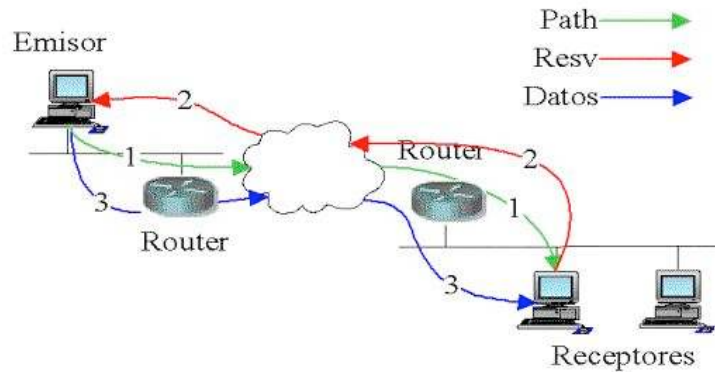


Figura 20: Uso de los mensajes Path y Resv

5.2. Flujos de datos

Existen tres conceptos básicos asociados con los flujos de datos que maneja el protocolo [17]:

Sesión RSVP. Es un flujo de datos identificado por su destino y por un protocolo de transporte particular. Sus componentes son los siguientes:

- Dirección IP destino. Dirección IP destino de los paquetes (unicast o multicast).
- Identificador del protocolo IP.
- Puerto destino (opcional).

Descriptor de flujo. Se llama así a una petición de reserva realizada por un sistema final. Está compuesto por los siguientes elementos:

Flowspec Especifica la calidad de servicio deseada. Incluye:

- Service class. Clase de servicio.
- Y dos parámetros numéricos: Rspec, que define la QoS deseada (Reserve) y Tspec, que describe el flujo de datos (Traffic).

Filter spec Designa un conjunto arbitrario de paquetes dentro de una sesión a los que aplicar la QoS definida por el flowspec. El formato depende de si se utiliza IPv4 o IPv6, pero básicamente es el siguiente:

- Dirección IP fuente + puerto UDP/TCP fuente.

5.3. Mensajes RSVP

Un mensaje RSVP está formado por una cabecera común, seguida de varios objetos de longitud variable.

5.3.1. Formato de la cabecera

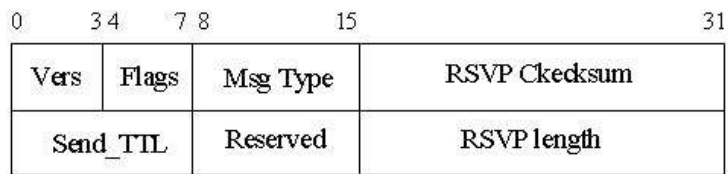


Figura 21: Cabecera de los mensajes RSVP

5.3.2. Campos de la cabecera

Vers Versión del protocolo RSVP. Actualmente la 1.

Flags No definido.

Msg Type Tipo de mensaje. Se enumeran a continuación:

- Path
- Resv
- Path_Err
- Resv_Err
- PathTear
- ResvTear
- ResvConf

RSVP Checksum Campo de verificación.

Send_TTL Indica el tiempo de vida (Time To Live) del mensaje.

RSVP Length Longitud total del mensaje expresada en bytes, incluyendo la cabecera y el cuerpo.

5.3.3. Formato de los objetos

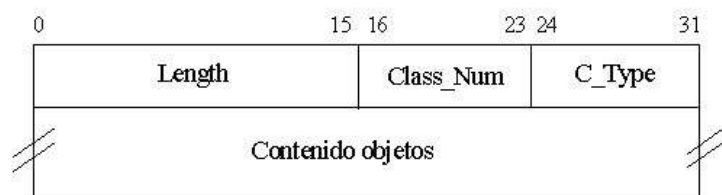


Figura 22: Formato de los objetos

Los campos de los objetos se explican a continuación:

Length. Longitud total del objeto expresada en bytes. Su valor debe ser siempre múltiplo de cuatro.

Class_Num Identifica la clase del objeto. Todas las implementaciones de RSVP reconocen las siguientes clases:

- NULL
- SESSION
- RSVP_HOP
- TIME_VALUES

- STYLE
- FLOWSPEC
- FILTER_SPEC
- SENDER_TEMPLATE
- SENDER_TSPEC
- ADSPEC
- ERROR_SPEC
- POLICY_DATA
- INTEGRITY
- SCOPE
- RESV_CONFIRM

C_Type Tipo de objeto. Identifica el tipo de objeto dentro de la clase.

5.3.4. Funcionamiento de RSVP

La fuente envía un mensaje Path a los destinos. Dicho mensaje se manda a una dirección que es una dirección de sesión, la cual podrá ser una dirección unicast o multicast. Cuando el destino reciba el mensaje Path podrá enviar un mensaje Resv a la fuente, que viajará justo por el camino inverso del mensaje Path. Dicho mensaje Resv identificará la sesión para la que se quiere hacer la reserva y el mensaje será reenviado hacia la fuente por los encaminadores. Finalmente, éstos reservarán los recursos necesarios analizando dicho mensaje.

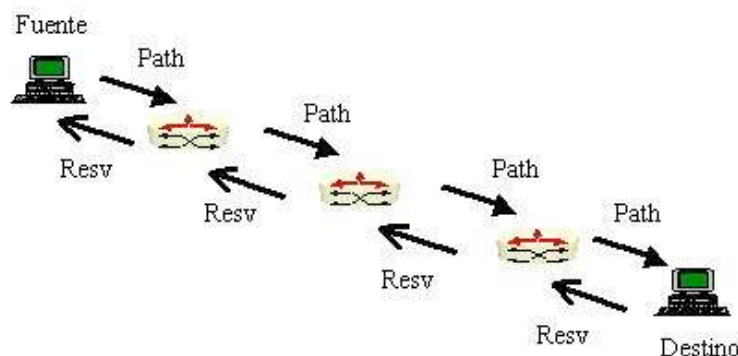


Figura 23: Funcionamiento de RSVP

Como vimos anteriormente, RSVP es un protocolo simplex. Los encaminadores reconocerán los paquetes pertenecientes a un flujo examinando la dirección origen y destino, el puerto origen y destino y el número de protocolo (p. ej. UDP). Puesto que RSVP es un protocolo de estado blando, se deberán mandar periódicamente mensajes Path y Resv para refrescar el estado.

5.4. RSVP-TE. Extensiones de RSVP para túneles LSP

RSVP-TE define los siguientes objetos extendidos para poder usarse con RSVP [18]:

- Objeto Etiqueta
- Objeto Petición de etiqueta
- Objeto Ruta Explícita
- Objeto Registrar Ruta

- Objeto Sesión LSP_TUNEL_IPv4
- Objeto Sesión LSP_TUNEL_IPv6
- Objeto Plantilla Emisor LSP_TUNEL_IPv4
- Objeto Plantilla Emisor LSP_TUNEL_IPv6
- Objeto Especificación Filtro LSP_TUNEL_IPv4
- Objeto Especificación Filtro LSP_TUNEL_IPv6
- Objeto Atributo Sesión
- Objetos TSPEC y FLOWSPEC para clases de servicio
- Objetos Hello

Se puede utilizar RSVP para establecer los LSP usando la distribución de etiquetas río abajo por demanda. Para establecer un LSP, el LSR de entrada mandará un mensaje Path. Dicho mensaje tendrá un objeto de petición de etiqueta y un objeto de sesión LSP_TUNEL_IPv4 o LSP_TUNEL_IPv6. Si un nodo no es capaz de realizar una asociación de etiquetas, mandará un mensaje PathErr con un error del tipo "Clase de Objeto Desconocido".

Cuando el mensaje Path llegue al LSR de salida, éste responderá con un mensaje Resv. Dicho mensaje contendrá el objeto etiqueta, utilizado como se describe a continuación:

El LSR de salida realizará una asociación de etiquetas e incluirá esta etiqueta en el objeto etiqueta, seguidamente mandará el mensaje río arriba. Cuando el siguiente LSR reciba este mensaje sabrá que la etiqueta incluida en el objeto etiqueta será la que debe usar como etiqueta de salida para ese flujo. Una vez hecho esto, el LSR asignará una etiqueta (que será la futura etiqueta entrante), la insertará en el objeto etiqueta y enviará el mensaje Resv río arriba. Este proceso se repetirá hasta que el mensaje llegue a la fuente. En ese momento se podrá decir que se ha establecido el LSP.

La siguiente figura muestra este proceso:

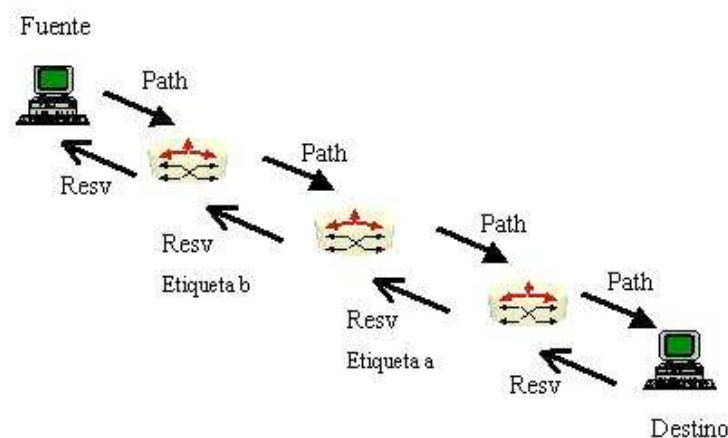


Figura 24: Establecimiento de túneles LSP mediante RSVP

El último LSR de la figura asigna la etiqueta a y la distribuye al LSR del centro. Éste LSR asigna la etiqueta b y la distribuye al LSR de entrada. El LSP para este flujo ya está creado.

Un LSR de entrada puede crear una ruta explícita. Esto se consigue añadiendo al mensaje Path el objeto ruta explícita. Este objeto encapsula una concatenación de saltos que constituyen el camino explícito, que puede ser especificado por un administrador o generarse automáticamente en base a una política determinada y una QoS requerida. Cuando un mensaje Path contiene el objeto de enrutamiento explícito, cada LSR reenviará el mensaje por el camino que dicho objeto especifique.

Una de las mayores ventajas del hecho de usar RSVP para establecer túneles LSP es que permite la asignación de recursos a través del camino. Pero no es obligatorio realizar la reserva de recursos cuando se establece el LSP, ya que se puede establecer un LSP sin reservar ningún tipo de recurso.

Por tanto, RSVP-TE es un protocolo específico para realizar la reserva de recursos en redes IP. Tradicionalmente no ha dado demasiados buenos resultados debido al carácter no orientado a la conexión del protocolo IP. Sin embargo, los LSP de MPLS sí que son orientados a conexión, por lo que usar RSVP-TE para el establecimiento de los mismos permite, simultáneamente, reservar ancho de banda para el tráfico que va a transcurrir por él.

RSVP-TE está muy implantado y estudiado. Eso significa que los esfuerzos e inversiones realizados en él pueden ser reutilizados. Además, se evita mantener dos conjuntos de protocolos en los LSR y LER, lo cual añadiría complejidad y coste. Para finalizar, decir que RSVP con las extensiones necesarias para ser usado con MPLS es una buena opción.

6. PCE

Aunque BGP es un protocolo de encaminamiento interdominio, no se trata de un protocolo adecuado para el funcionamiento en conjunto con MPLS, ya que no presenta facilidades para aplicar ingeniería de tráfico, no utiliza las métricas habituales de los IGP, sino políticas, y, además, no sigue la misma filosofía de MPLS, sino que sigue las ideas de IP.

Para ello, se buscan constantemente modos de establecer LSP entre dominios MPLS adyacentes de forma sencilla. Se han presentado propuestas, como que RSVP-TE pueda saltar la frontera de un AS y algunas otras, pero esto resulta aún insuficiente. En este tema, el IETF está desarrollando una arquitectura que permita el encaminamiento del tráfico interdominio para GMPLS, denominada PCE (*Path Computation Element*) [19].

La idea es que PCE pueda liberar a los nodos tanto de las redes MPLS como GMPLS de las tareas de cálculo de caminos, puesto que hoy en día es algo complejo, ya que no sólo se pretende que el tráfico llegue a su destino, sino que además lo haga con seguridad, fiabilidad, garantías y optimizando los recursos de la red sobre la que viaja.

Los elementos que entran en juego en esta arquitectura se explican a continuación:

- PCE** Es el elemento encargado de calcular caminos usando para ello restricciones y funciones objetivo. Por tanto, este tipo de dispositivos se encuentran especializados en el cálculo de caminos y son a éstos a los que todos los nodos de la red pueden dirigir sus peticiones y que lleven el trabajo pesado de esta tarea.
- TED** Se trata de una base de datos de ingeniería de tráfico que conoce la topología de la red.
- PCC** Es el cliente que solicita los servicios de PCE para que le proporcione un camino con unas características concretas.
- PCECP** (*Path Computation Element Communication Protocol*). Es el protocolo de comunicaciones que se emplea entre el PCC y el PCE o entre varios PCE para colaborar y calcular el camino hacia el destino.

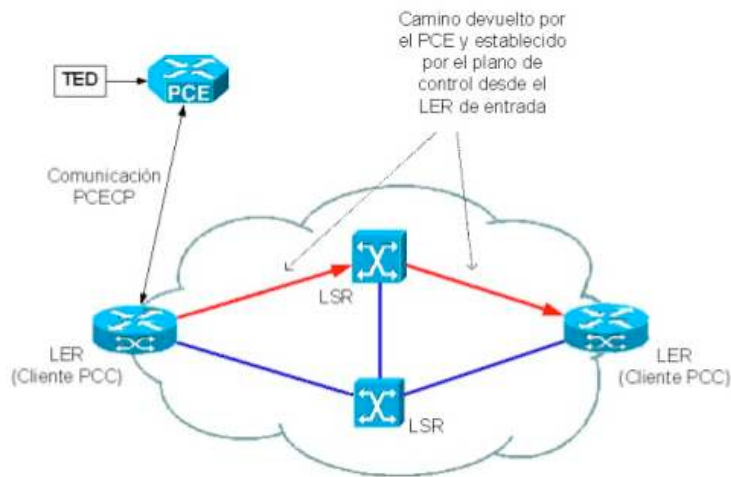


Figura 25: Componentes típicos de una arquitectura PCE

De este modo, puede haber un PCE que calcule él solo el camino completo, varios PCE que hagan lo propio con la consiguiente liberación de carga, o varios PCE que colaboren de forma que puedan calcular cada uno solamente un segmento del camino.

6.1. Funcionamiento de PCE

Desde el principio, la arquitectura PCE se diseñó para funcionar tanto en entornos intradominio como en entornos interdominio y, en este último caso, el funcionamiento que se lleva a cabo es el siguiente:

- Los PCE llevan asociados una base de datos de ingeniería de tráfico denominada TED, similar a la LSD en OSPF-TE, que está alimentada por los protocolos de encaminamiento interior utilizados (OSPF-TE, ISIS-TE...).
- Inicialmente llega a un dominio MPLS un tipo de tráfico no MPLS. Seguidamente, el LER de entrada debe empezar el establecimiento de un camino con unas determinadas restricciones para llegar al destino. Para ello, y a través de su módulo PCC, pide al PCE de su dominio que calcule el mejor camino según las necesidades impuestas y utiliza para ello PCECP. Después de esto, el PCC queda en espera.
- En este momento, el PCE empieza a calcular el camino. En el caso de que los dos extremos de la comunicación se encuentren en el mismo AS, dicho PCE calculará el camino de forma autónoma o ayudándose de otros PCE, colaborando entre ellos. Si por el contrario el origen y el destino no se encuentran en el mismo AS, el PCE debe comunicarse obligatoriamente con otro PCE del dominio destino mediante PCECP para calcular el camino entre ambos, ya que su base de datos (TED) solamente contiene información sobre el dominio al que pertenece dicho PCE y no la de otros.
- Una vez que el PCE haya calculado el camino y se lo haya devuelto al PCC del LER de entrada utilizando el protocolo PCECP, este último emplea los protocolos de señalización habituales como RSVP-TE para establecer el camino y reservar el recurso (PCE no establece caminos, únicamente los calcula).
- Por último, y con todo lo anterior realizado, el tráfico comienza a fluir por el camino establecido. Transcurrido un tiempo, los IGP actualizan la TED de los distintos AS con los posibles cambios en la topología de la red, para que el PCE en cuestión conozca el cambio de la disponibilidad de los recursos para calcular los caminos que se le soliciten con la mayor información posible.

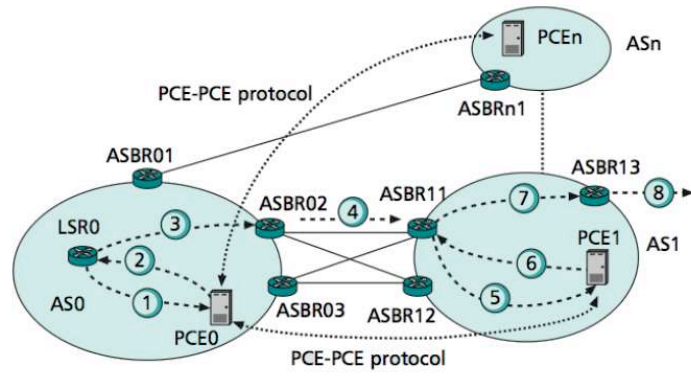


Figura 26: Ejemplo de funcionamiento de la arquitectura PCE

En la arquitectura PCE, la verdadera dificultad reside en el hecho de que probablemente un nodo PCE no sea capaz de calcular el camino completo por sí mismo, sino que tenga que colaborar con otros. En este caso se desencadena un mecanismo para coordinar todo el proceso, en el que el PCC no tiene constancia de ello, únicamente recibe el camino.

Parte II

Simulaciones e investigación de cuestiones propuestas

7. Búsqueda de entornos de simulación

Una vez familiarizados con todas las tecnologías interrelacionadas entre sí y con el objeto de estudio de este proyecto, el principal objetivo ha sido la realización de simulaciones que permitiesen sentar una base teórica y práctica desde la cual comenzar a desarrollar los principales trabajos propuestos para esta beca. Para ello, y puesto que se desea probar una nueva arquitectura para el encaminamiento interdominio como es PCE, pero que hace uso de información relativa al intradominio, las características deseables en el simulador a utilizar para poder llevar a cabo las pruebas son las siguientes:

- Soporte de OSPF-TE.
- Soporte de ISIS-TE.
- Soporte de MPLS-TE intradominio.
- Soporte de MPLS-TE interdominio.
- Soporte de BGP.
- Soporte de PCE intradominio.
- Soporte de PCE interárea.
- Soporte de PCE interdominio.
- Soporte de PCECP.
- Soporte de RSVP-TE intradominio.
- Soporte de RSVP-TE interdominio.
- Soporte de IPv4.
- Soporte de IPv6.
- Soporte del trabajo conjunto de todos estos protocolos.

Además de todo esto, resultaría muy importante de cara a la investigación y desarrollo de proyectos futuros que se permita la extensión de los protocolos anteriormente citados. De esta manera, tendríamos más libertad de movimiento a la hora de desarrollar las investigaciones relativas a los principales objetivos marcados por esta beca. Por tanto, sería muy recomendable que el simulador facilitase las siguientes tareas:

- Añadir extensiones a OSPF-TE.
- Añadir extensiones a ISIS-TE.
- Añadir extensiones a BGP.
- Añadir extensiones a PCECP.
- Modificar el funcionamiento de los protocolos anteriores para que utilicen las nuevas extensiones.
- Debe permitir que se añadan elementos de red completamente nuevos y que puedan interaccionar con los existentes.

Bajo todas estas suposiciones y requisitos deseables, se pasó a buscar la suite de simulación que servirá como base para todas las pruebas a realizar.

Han sido varios los entornos de simulación estudiados, pero la mayor parte de ellos fueron descartados rápidamente por no disponer de la mayoría de los requisitos anteriormente enumerados. Sin embargo, ha habido dos que merece la pena destacar por haber sido dos firmes candidatos. Éstos son OPNET Modeler y TOTEM Project, y aunque finalmente no se eligiese ninguno de los dos, a continuación mostraremos los resultados de los estudios realizados sobre cada uno de ellos y los resultados e impresiones obtenidos.

7.1. OPNET Modeler

Modeler [20] es la herramienta diseñada por OPNET para el modelado y simulación de redes, basada en la teoría de las *redes de colas*². Incorpora librerías para facilitar el modelado de las redes de comunicación. Originalmente fue desarrollado por el MIT e introducido al mercado en 1987 como el primer simulador comercial, manteniéndose como referente mundial en el modelado de redes simuladas hasta la actualidad.

En primer lugar, se realizaron búsquedas en la web de OPNET [20] para comprobar las capacidades y limitaciones que presenta Modeler. En el sitio web dedicado al producto, la información referente parece bastante específica y clara, al menos en lo referente a los protocolos y tecnologías que implementa. Pueden consultarse dichos datos en la dirección http://www.opnet.com/support/des_model_library/index.html, donde se puede encontrar el listado de protocolos a los que Modeler da soporte en primera instancia gracias al uso de la librería DES, *Discrete Event Simulation Model Library* [21].

Una vez realizada la primera búsqueda en el portal web del proveedor, y al comprobar que el producto no daba soporte en primera instancia a algunos protocolos y tecnologías de vital importancia para el caso que nos concierne (como PCE, PCECP o la propia extensión de Ingeniería de Tráfico en algunos protocolos), se encaminó la búsqueda hacia sitios más generales relacionados con el mundo académico para averiguar si la aplicación daba soporte de alguna manera a esas tecnologías que no aparecían en la lista oficial. Desafortunadamente, y en parte por tratarse de tecnologías relativamente nuevas, no se encontraron implementaciones específicas orientadas a dichas tecnologías. Sin embargo, sí se hallaron varios proyectos en los que se dejaba constancia de que Modeler es una herramienta que permite la extensión de sus protocolos gracias al uso de la librería *Discrete Event Simulation Model*, en conjunto con otras creadas por el usuario.

En definitiva, y aunque no se hayan encontrado documentos o fuentes que simulen la arquitectura PCE sobre OPNET Modeler, parece en un principio que no existe ningún inconveniente que impida que la herramienta tenga la capacidad de ser extendida siguiendo esta línea de investigación, por lo tanto se consideró como posible software de simulación.

7.1.1. Estudio comparativo de OPNET Modeler

El resultado final de esta labor de investigación concluye que OPNET Modeler da soporte a las siguientes tecnologías y soportes relacionadas con la arquitectura interdominio que es objeto de nuestro estudio, *Path Computation Element*:

²Una red de colas es un sistema donde existen varias colas y los trabajos van fluyendo de una a otra según ciertos criterios preestablecidos.

	OPNET Modeler	
		¿TE?
OSPF-TE	Sí	Sí
ISIS-TE	Sí	Sí
MPLS-TE intradominio	Sí	?
MPLS-TE interdominio	?	?
BGP	Sí	
PCE intradominio	?	
PCE interdominio	?	
PCECP	?	
RSVP-TE intradominio	Sí	Sí
RSVP-TE interdominio	?	?
IPv4	Sí	
IPv6	Sí	

Cuadro 1: Estudio comparativo de OPNET Modeler

Una vez analizada la tabla comparativa con los resultados obtenidos, viendo que OPNET Modeler no satisfacía la mayoría de los requisitos necesarios para realizar simulaciones en un escenario con varios PCE's, y ante el hecho de ser un software privativo (sólo se podría conseguir mediante la realización de un convenio entre la UEx y OPNET para la adquisición de licencias para usos académicos y de investigación, y por diversas circunstancias dicho acuerdo no se ha producido, puesto que OPNET pensó que su uso para el apoyo de becas de CISCO Systems podría tener un uso comercial, en vez de meramente académico y de investigación), se decidió buscar otras alternativas dentro del mundo del software libre.

7.2. TOTEM Project

Como punto de partida, cabe destacar que esta herramienta es de libre distribución (publicada bajo la licencia GPL 2.0 en la versión 3.2 de este software), por lo que en un principio parece que se encontrará más soporte y ayuda que para otros simuladores de carácter privativo. No obstante, el Proyecto TOTEM [22] comenzó en 2003 y tuvo una duración de 3 años, disolviéndose o congelándose en 2006. Sin embargo, el equipo de desarrolladores de la herramienta han seguido trabajando en ella y lanzando nuevas versiones, siendo la actual versión 3.2 liberada en noviembre de 2007.

En primer lugar, se realizó un análisis exhaustivo del sitio web del Proyecto TOTEM. En dicho análisis, además de adquirir información sobre los objetivos y principios del proyecto, se estudiaron las distintas herramientas implementadas por los miembros de este proyecto, y de manera más profunda la más interesante para nuestro propósito, TOTEM Toolbox [23], de libre distribución. Dicha herramienta fue descargada, y tras varios problemas iniciales por los cuales no se ejecutaba en Mac OS X, se descargó el código fuente para recompilarla. Una vez realizada esta reconstrucción del simulador, se pudo ejecutar sin ningún tipo de problema.

Seguidamente, y al comprobar que existían varias carencias en la página web dedicada al conjunto de herramientas de simulación TOTEM, se decidió contactar a través de correo electrónico con uno de los principales responsables del Proyecto TOTEM, el profesor Olivier Bonaventure, de la Université Catholique de Louvain, en Bélgica. En estos correos, el profesor Bonaventure nos remitió a la profesora Cristel Pelsser, también de la

Université Catholique de Louvain, ya que su tesis doctoral, *Interdomain traffic engineering with MPLS*³, trataba ampliamente sobre PCE.

Una vez obtenida la forma de contactar con la Doctora Pelsser, se contactó con ella y se le hizo llegar, nuevamente vía e-mail, una serie de preguntas sobre las capacidades y limitaciones de TOTEM en cuanto a PCE se refiere, haciendo hincapié en los problemas que pudo encontrar a la hora de utilizarlo para su tesis doctoral. Aunque la respuesta se hizo esperar, finalmente la Doctora Pelsser aclaró que TOTEM Toolbox no soporta de manera nativa PCE y PCECP. No obstante, indicó que el simulador en sí podría verse como una abstracción de un PCE, ya que podría calcular determinadas rutas dada una serie de restricciones. Sin embargo, el gran problema reside a la hora de comunicar PCC's con PCE's, ya que el protocolo especificado para ello, PCECP, no está implementado en TOTEM.

Llegados a este punto, parece que existe una limitación considerable en lo referente a la utilización de este simulador como base para los estudios sobre PCE en este proyecto, ya que no existe ninguna solución nativa que implemente de manera fiable el comportamiento estándar de un PCE y el protocolo de comunicación de los PCC's con éste, el PCECP. No obstante, y nuevamente gracias a las comunicaciones con la Dr. Pelsser, se pudo averiguar que existe una “emulación” de los mensajes del protocolo PCECP entre los elementos que interactúan en la comunicación a través de un formato de mensajes diseñado específicamente para este propósito por Gael Monfort. Estos mensajes son enviados y recibidos a través de sockets habilitados para ello en el simulador TOTEM.

En un nuevo e-mail se le preguntó a la Dr. Pelsser y al Sr. Monfort dónde se podría encontrar una especificación del formato de estos mensajes alternativos al PCECP, y cómo integrarlos en el simulador, aunque aún no se ha recibido respuesta alguna.

7.2.1. Características disponibles en TOTEM

Como ya se comentó anteriormente, TOTEM Toolbox es un software de código abierto, distribuido bajo una licencia GPL de libre distribución. Por tanto, parece que uno de los requisitos fundamentales que se le exigían a esta herramienta se cumple, es decir, es un software de simulación extensible con un conjunto de algoritmos de emulación de diversos protocolos también extensible. A continuación se listan algunos de los algoritmos que incluye la herramienta y que son interesantes para nuestro propósito:

DAMOTE (*Decentralized Agent for MPLS Online Traffic Engineering*): Introduce básicamente dos funcionalidades, establecimiento de LSP's en arquitecturas Diffserv basados en QoS y copia de seguridad local y global de las rutas LSP's para una rápida restauración del sistema en caso de sobrecarga o caída.

MIRA y SAMCRA Estos algoritmos son usados para calcular LSP's entre dos nodos de la red.

CBGP Este algoritmo implementa una solución para los enrutamientos BGP.

IGP-WO (*Interior Gateway Protocol-Weight Optimization*): El objetivo de este módulo es encontrar un enlace dentro del dominio de la red para realizar un balance óptimo de carga en la red.

SAMTE (*Scalable Approach for MPLS Traffic Engineering*): SAMTE se utiliza para encontrar el menor número de LSP para establecer en la red y que ésta sea factible.

Partiendo de esta base sobre la filosofía del simulador y los algoritmos que dispone, pasemos a continuación a enumerar los requisitos que TOTEM Toolbox cumple en su versión “estándar” o sin modificar:

³Este documento puede encontrarse en <http://inl.info.ucl.ac.be/publications/interdomain-traffic-engineering-mpls>.

	TOTEM Toolbox	
	Soportado	¿Algoritmo/ Extensión?
OSPF-TE	Sí	Algoritmo DAMOTE
ISIS-TE	Sí	DAMOTE
MPLS-TE intradominio	Sí	MIRA, SAMCRA y SAMTE
MPLS-TE interdominio	?	?
BGP	Sí	CBGP
PCE intradominio	Sí	Abstracción de la herramienta (Pelsser)
PCE interdominio	Sí	Abstracción de la herramienta (Pelsser)
PCECP	No	Mensajes no estandarizados a través de sockets (Monfort)
RSVP-TE intradominio	Sí	Incluido en la herramienta
RSVP-TE interdominio	?	?
IPv4	Sí	Incluido en la herramienta
IPv6	Sí	?

Cuadro 2: Estudio comparativo de TOTEM Toolbox

Como se puede comprobar, el único problema lo encontramos con las extensiones interdominio de MPLS y RSVP, que no quedan muy definidas en la documentación consultada, y con todo lo referente al objeto de estudio de este proyecto, PCE y su protocolo PCECP. Sin embargo, en este punto, y atendiendo al correo recibido por parte de la Dr. Pelsser, parece que existiría una forma de realizar simulaciones de escenarios con elementos PCE, aunque ésta no sería muy ortodoxa ni se acercaría a los pocos estándares existentes en la actualidad referentes a PCE y PCECP.

Una vez recopilada toda esta información y con los resultados obtenidos, parece que TOTEM Toolbox podría haber sido nuestra herramienta de simulación base. Sin embargo, el hecho de tener que emular los mensajes del protocolo PCECP de una manera tan poco *cuidada* nos hizo replantearnos la elección de TOTEM Toolbox como suite de simulación.

Llegados a este punto, y ante la inexistencia en el mercado de algún simulador que implemente de manera nativa la arquitectura PCE y su protocolo PCECP, parece que nos encontramos en un callejón sin salida. Sin embargo, en el grupo de investigación GÍTACA se está elaborando un simulador⁴ de la arquitectura PCE en entornos interdominio, y parte de la base teórica en la que se basa dicho simulador es el estudio previo de las tecnologías realizado en este proyecto. Además, se ha colaborado en algunos aspectos durante su desarrollo, concretamente en la definición y validación de los escenarios con las topologías a simular en XML y Java.

⁴Se puede encontrar más información acerca de este proyecto en la web <http://patanegra.unex.es/opensimripca>.



Figura 27: OpenSimRIPCA

8. Colaboración con el proyecto OpenSimRIPCA

Como se comentó en el punto anterior, durante una parte de la duración de este proyecto se colaboró con en el grupo de investigación GÍTACA en la elaboración de un simulador que implemente la arquitectura PCE en entornos interdominio, concretamente en la definición y validación de los escenarios con las topologías a simular en XML y Java. En este apartado se analizarán algunos de los contenidos teóricos requeridos para esta colaboración.

8.1. Validación XML

La Validación XML (*eXtensible Markup Language*) es la comprobación de que un documento en lenguaje XML está bien formado y se ajusta a una estructura definida. Un documento bien formado sigue las reglas básicas de XML establecidas para el diseño de documentos. Un documento válido además respeta las normas dictadas por su DTD (definición de tipo de documento) o esquema utilizado (XML Schema).

En primer lugar, los documentos XML deben basarse en la sintaxis definida en la especificación XML para ser correctos (documentos bien formados). Esta sintaxis impone cosas como la coincidencia de mayúsculas/minúsculas en los nombres de etiqueta, comillas obligatorias para los valores de atributo, etc. Sin embargo, para tener un control más preciso sobre el contenido de los documentos es necesario un proceso de análisis más exhaustivo.

La validación es la parte más importante dentro de este análisis, ya que determina si un documento creado se ciñe a las restricciones descritas en el esquema utilizado para su construcción. Controlar el diseño de documentos a través de esquemas aumenta su grado de fiabilidad, consistencia y precisión, facilitando su intercambio entre aplicaciones y usuarios. Cuando creamos documentos XML válidos aumentamos su funcionalidad y utilidad.

8.1.1. Parsers XML

El *parser*, procesador o analizador sintáctico de XML es la herramienta principal de cualquier aplicación XML. Mediante el parser no solamente se comprueba si los documentos XML están bien formados o si son válidos, sino que también se pueden incorporar a una aplicación, dotando a éstas la capacidad de manipular y trabajar con documentos XML.

El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis, y capturan la jerarquía implícita de la entrada. Un analizador léxico crea tokens

de una secuencia de caracteres de entrada y son estos tokens los que son procesados por el analizador sintáctico para construir la estructura de datos, por ejemplo un árbol de análisis o árboles abstractos de sintaxis.

Podemos dividir los parsers XML en dos grupos principales:

- **Sin validación:** el parser no valida el documento utilizando un DTD o un XML Schema, sino que sólo chequea que el documento esté bien formado de acuerdo a las reglas de sintaxis de XML (sólo hay una etiqueta raíz, las etiquetas están cerradas, etc.).
- **Con validación:** además de comprobar que el documento está bien formado según las reglas anteriores, comprueba el documento utilizando un DTD o un XML Schema (ya sean internos o externos).

Para el desarrollo de esta colaboración se han utilizado los parsers incluidos en el *Java Development Kit* y el parser del proyecto Xerces⁵.

8.2. API's de XML para Java

Aunque existe una infinidad de librerías y API's para la manipulación de XML en Java como son JAXP y JDOM, en este proyecto se ha tratado únicamente con dos, DOM y SAX.

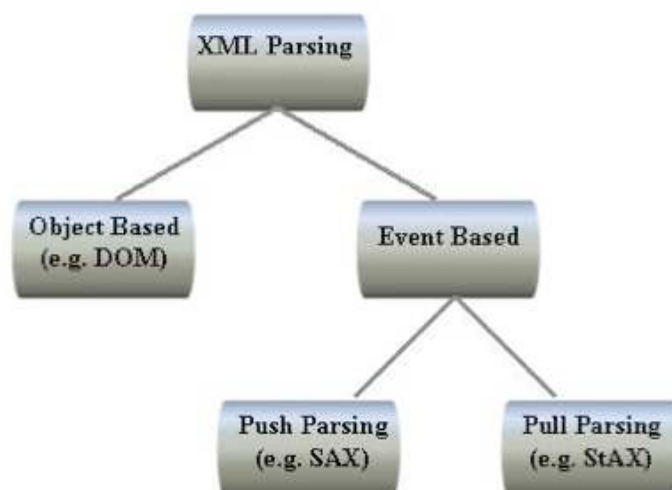


Figura 28: Tipos de procesamiento XML

8.2.1. DOM

El *Document Object Model* (DOM) es esencialmente un modelo computacional diseñado por el consorcio W3C (*World Wide Web Consortium*), a través del cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML. Su objetivo es ofrecer un modelo orientado a objetos para el tratamiento y manipulación en tiempo real (o de forma dinámica) a la vez que de manera estática de páginas de internet.

En nuestro caso, el DOM es una API para acceder, añadir y cambiar dinámicamente contenido estructurado de documentos XML con lenguajes como C++, PHP o el propio Java.

En Java, el funcionamiento de DOM es muy sencillo, pues genera un árbol jerárquico en memoria del documento o información en XML, basándose en cada elemento $\langle Etiqueta1 \rangle$, $\langle Etiqueta2 \rangle$, $\langle Etiqueta3 \rangle$, etc., que es considerado un nodo dentro del árbol. Gracias a esta estructuración, el *parser* puede trabajar con la información almacenada de manera transparente.

Las ventajas de utilizar DOM como API/Modelo de objetos serían las siguientes:

⁵Disponible en <http://xerces.apache.org/xerces-j/>

- Puede ser agregado un nodo (información) en cualquier punto del árbol.
- Se puede eliminar información de un nodo en cualquier punto del árbol.
- Lo anterior se ejecuta sin incurrir en las penalidades o limitaciones de manipular un archivo de alguna otra manera.

8.2.2. SAX

El *Simple API for XML* (SAX) es una API de XML con un parser de acceso secuencial propio, diseñada para proporcionar un mecanismo de lectura de datos en documentos XML basado en eventos. Surgió como alternativa al modelo DOM, y aunque no hay una especificación formal que la defina, actualmente se toma como una norma o estándar.

A diferencia de DOM, que genera un árbol jerárquico en memoria, SAX procesa la información en XML conforme ésta sea presentada (evento por evento), manipulando cada elemento a un determinado tiempo, sin incurrir en un uso excesivo de memoria.

La utilización de SAX como API de desarrollo en XML brinda las siguientes ventajas:

- SAX es un parser ideal para manipular archivos de gran tamaño, ya que no requiere generar un árbol en memoria como es el caso de DOM.
- Es más rápido y sencillo que utilizar DOM.

La sencillez antes mencionada tiene su precio, puesto que SAX funciona por eventos y no es posible manipular información una vez procesada. En DOM no existe esta limitación, ya que se genera el árbol jerárquico en memoria y es posible recorrer la estructura de nuevo en cualquier momento.

8.3. Aportación a OpenSimRIPCA

Como se comentó anteriormente, se ha colaborado con en el grupo de investigación GÍTACA en la elaboración de un simulador que implemente la arquitectura PCE en entornos interdominio. Dicha colaboración se ha basado concretamente en la definición y validación de escenarios con las topologías a simular en XML y Java. Por tanto, y utilizando todos los conocimientos adquiridos en los puntos anteriores, se diseñaron varias clases, descritas a continuación:

ValidarXML Esta clase permite, mediante la API SAX, validar un fichero o documento XML con la implementación de un escenario frente a un esquema XSD con la especificación de un modelo bien formado y correcto de escenario válido para OpenSimRIPCA.

ProcesadorXML Esta clase permite procesar un documento XML con un escenario de OpenSimRIPCA para extraer su información y poder almacenarla en clases Java, cada una de ellas con las características de cada uno de los elementos que intervienen en una simulación. Utiliza la API DOM.

9. Relaciones entre OSPF-TE, BGP y PCE

Evidentemente, la arquitectura PCE sería el punto de unión entre el encaminamiento intradominio y el interdominio, ya que para eso fue diseñada. Ahora quedaría ver cómo completan la información de las TED de los PCE cada uno de los dos protocolos, OSPF-TE y BGP.

9.1. OSPF-TE y PCE

Sabemos de antemano que las bases de datos de ingeniería de tráfico (TED) de PCE serían rellenas con los datos de encaminamiento calculados por OSPF-TE [24], ya que al ser este último un protocolo de encaminamiento interior y al estar los PCE dentro de un AS, los propios enrutadores internos del AS deberán saber qué camino deben seguir los paquetes enviados desde un nodo origen hasta otro destino dentro del sistema. Haciendo un símil, podríamos tomar las LSD de OSPF-TE como las TED de PCE en cuanto al encaminamiento interior se refiere.

Sin embargo, el problema se presenta a la hora de hacer un encaminamiento interdomino. Está claro que, en el caso ideal de que la arquitectura PCE se implantase en la red, tendría que haber un mecanismo para que, durante el periodo de implantación y adaptación de los sistemas, las TED de los PCE de un determinado AS obtuviesen información por parte de los nodos BGP sobre qué camino deben seguir los paquetes que salgan de ese sistema para llegar a otro sistema remoto.

9.2. BGP y PCE

Quizá la manera más transparente para BGP sería considerar al PCE como otro nodo BGP más, por lo que se tendría que desarrollar un nuevo conjunto de mensajes, protocolos de intercambio de información, etc. que hiciesen posible la sincronización de las TED con las tablas de ruta de los nodos BGP. Desde luego, esto supondría obviar por completo la filosofía restrictiva de BGP, ya que los PCE tendrían de esta manera información detallada sobre la topología de sus AS vecinos. No obstante, y al no disponer nuevamente de un software de simulación que nos permitiese realizar pruebas bajo estas bases, esta idea no deja de ser una mera hipótesis.

No obstante, K. Kumaki y T. Murai proponen una extensión al protocolo BGP para introducir la capacidad de descubrimiento de PCEs en determinadas redes [25]. Para ello, introducen un nuevo atributo en los mensajes de adquisición de vecinos de BGP, llamado *PCE Discovery Attribute*.

9.2.1. PCE Discovery Attribute

La información referente al PCE es transportada en el campo *Path Attributes* de los mensajes UPDATE de BGP [13]. Los bits bandera incluidos en la cabecera de estos mensajes se establecerán de la siguiente manera:

- El bit *Optional* se activa (1).
- El bit *Transitive* se desactiva (0).
- El bit *Partial* se desactiva (0).
- El bit *Extended Length* se activa (1).

Por otro lado, el campo *Path Attributes* se codificará de la siguiente manera: $\langle \text{Longitud}, \text{Lista de TLVs} \rangle$, donde:

Longitud : Es la longitud en bytes de la lista de TLVs transportada en el campo *Path Attribute*. Su tamaño es de dos octetos.

Lista de TLV's : Contiene una lista de TLVs, cada uno de los cuales puede ser un PCE descubierto. Su tamaño es variable.

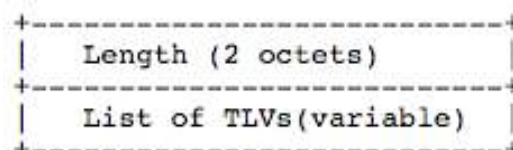


Figura 29: Codificación del campo *Path Attributes*

9.2.2. Funcionamiento del PCE Discovery Attribute

Con la extensión del *PCE Discover Attribute*, BGP funcionaría tal como se explica a continuación para encontrar y actualizar información de encaminamiento con un PCE:

- **Proceso de Transmisión:** Los nodos BGP anuncian la dirección del nodo PCE de su sistema. Esta dirección irá incluida en el *Path Attribute* del mensaje de adquisición de vecinos de BGP, encapsulada dentro del *PCE Discovery Attribute*.
- **Proceso de recepción:** El nodo BGP que recibe el *PCE Discovery Attribute* lo registra en su tabla de enrutamiento con su dirección correspondiente.
- **Procedimiento de cálculo de ruta ante una solicitud:** Si se recibe una solicitud de dirección de un PCE desde un proceso de cálculo de rutas (un PCC generalmente), BGP recupera la ruta de su tabla y devuelve la dirección incluida anteriormente en el *PCE Discovery Attribute*. Si el atributo no está en la tabla de enrutamiento del nodo BGP, se notifica un error de cálculo de rutas al PCC solicitante. Si existen dos o más direcciones de PCEs en la tabla del nodo BGP, se devuelven todas.

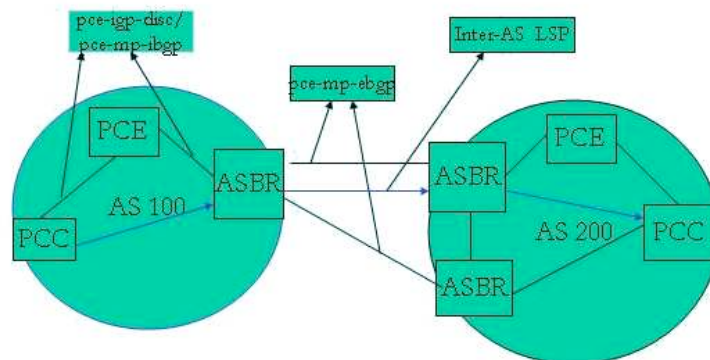


Figura 30: Esquema de comunicación BGP/PCE

Nuevamente, al no disponer de ningún entorno de simulación que se adapte a nuestras necesidades, no se ha podido realizar ningún tipo de simulación sobre este supuesto, por tanto tendremos que considerarlo todavía como una hipótesis también, aunque como se ha comentado anteriormente, el IETF está trabajando en esta propuesta, proponiendo nuevos *drafts*, como *BGP protocol extensions for Path Computation Element (PCE) Discovery in a BGP/MPLS IP-VPN* [25], y revisando continuamente diversos RFC.

10. Conclusiones y trabajo futuro

Llegados a este punto de la investigación, las bases teóricas del objeto de estudio de este proyecto han quedado bien definidas. Sin embargo, y tras haber pasado la mayor parte del tiempo buscando una suite de simulación que nos permitiese ahondar en los trabajos propuestos, no se ha encontrado ninguna herramienta que nos permitiese contrastar los datos de las investigaciones realizadas, así como para apoyar los resultados de las futuras. Por tanto, parece claro que es necesario un simulador competente para poder continuar con el propósito de este trabajo.

No obstante, el grupo de investigación GÍTACA está elaborando un simulador de la arquitectura PCE, con lo que esperamos que pueda servir como suite de simulación para escenarios que nos permitan desarrollar los trabajos propuestos que quedan por hacer. Aún así, también sería muy conveniente poder contar con alguno de los equipos que se detallaron en el documento de solicitud de esta beca para facilitar las tareas investigativas.

Parte III

Anexos

Glosario de términos

ABR	Area Border Router. Referente a OSPF
AS	Autonomous System
ASBR	Autonomous System Border Router. Referente a OSPF
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
CoS	Class Of Service
CSPF	Constrained Shortest Path First
DOM	Document Object Modeler
EGP	Exterior Gateway Protocol
FEC	Forward Equivalence Class. Referente a MPLS
GMPLS	Generalized Multiprotocol Label Switching
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet Protocol
IR	Interior Router. Referente a OSPF
IS-IS	Intermediate System to Intermediate System
ISIS-TE	Intermediate System to Intermediate System – Traffic Engineering
ISP	Internet Service Provider
LER	Layer Edge Router. Referente a MPLS
LS	Label Stack. Referente a MPLS
LSD	Link State Database. Referente a OSPF
LSP	Label Switched Path. Referente a MPLS
LSR	Label Switch Router. Referente a MPLS
MPLS	Multiprotocol Label Switching
MPLS-TE	Multiprotocol Label Switching – Traffic Engineering
OSI	Open System Interconnection
OSPF	Open Shortest Path First
OSPF-TE	Open Shortest Path First – Traffic Engineering
PCC	Path Computation Client
PCE	Path Computation Element
PCECP	Path Computation Element Communication Protocol
QoS	Quality Of Service

RIP	Routing Information Protocol
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
SAX	Simple API for XML
TCP	Transmission Control Protocol
TED	Traffic Engineering Database
TLV	Type Length Value
XML	eXtensible Markup Language

Bibliografía, artículos y enlaces

1. TOTEM Toolbox User Guide (PDF)
<http://gforge.info.ucl.ac.be/docman/view.php/28/3203/TOTEM-3.2-UserGuide.pdf>
2. Pelsser, C. *Interdomain Traffic Engineering with MPLS*.
3. A. Sprintson, M. Yannuzzi, A. Orda, X. Masip-Bruin. *Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks*.
4. M. Yannuzzi, X. Masip-Bruin, S. Sánchez, J. Domingo-Pascual. *On the Challenges of Establishing Disjoint QoS IP/MPLS Paths Across Multiple Domains*.
5. IBM developerWorks. *XML Schema Validation in Xerces-Java 2*
6. B. McLaughlin. *Java y XML*. Anaya Multimedia. (Ba-3198)
7. E. Burke. *Java y XSLT*. Anaya Multimedia. (Ba-3320)
8. *ELEMENT, The Path Computation Element Website*.
<http://pathcomputationelement.com/>
9. *Path Computation Element (PCE): IETF's hidden jewel*
<http://technologyinside.com/2007/04/10/path-computation-element-pce-ietf-hidden-jewel/>
10. CISCO Systems. *BGP Case Studies*.
<http://www.cisco.com/en/US/tech/tk365/technologies-tech-note09186a00800c95bb.shtml>
11. Apache XML Project. *Xerces Java Parser*.
<http://xerces.apache.org/xerces-j/>
12. Wikipedia.
<http://es.wikipedia.org>
13. Osmosis Latina. *DOM, SAX, JDOM y JAXP en XML*.
<http://www.osmosislatina.com/xml/domsax.htm>

Índice de figuras

1.	Sistema autónomo usando OSPF-TE	9
2.	Esquema de las funciones de reenvío y encaminamiento	11
3.	Formato del paquete	12
4.	Entradas de la tabla de encaminamiento	13
5.	Asociación de etiquetas río abajo	14
6.	Asociación de etiquetas río arriba	14
7.	Algoritmo de reenvío	15
8.	Situación de MPLS en el modelo de referencia OSI	16
9.	Tipos de nodos MPLS	18
10.	Río abajo solicitado	19
11.	Río abajo no solicitado	19
12.	Formato de las etiquetas	19
13.	Esquema MPLS con los distintos paquetes que entran en juego	20
14.	Ejemplo de pila de etiquetas	21
15.	Topología de red para explicar la ingeniería de tráfico	22
16.	Esquema general del funcionamiento de BGP	25
17.	Campo NLRI	27
18.	Ejemplo de distribución de etiquetas	28
19.	Situación de RSVP en la familia de protocolos TCP/IP	29
20.	Uso de los mensajes Path y Resv	30
21.	Cabecera de los mensajes RSVP	31
22.	Formato de los objetos	31
23.	Funcionamiento de RSVP	32
24.	Establecimiento de túneles LSP mediante RSVP	33
25.	Componentes típicos de una arquitectura PCE	35
26.	Ejemplo de funcionamiento de la arquitectura PCE	36
27.	OpenSimRIPCA	42
28.	Tipos de procesamiento XML	43
29.	Codificación del campo <i>Path Attributes</i>	45
30.	Esquema de comunicación BGP/PCE	46

Referencias

- [1] C. Hedrick. *Routing Information Protocol*. IETF RFC 1058. Junio de 1988.
- [2] G. Malkin. *RIP Version 2*. IETF RFC 2453. Noviembre de 1998.
- [3] J. Moy. *OSPF Version 2*. IETF RFC 2178. Abril de 1998.
- [4] D. Katz, K. Kompella. *Traffic Engineering (TE) Extensions to OSPF Version 2*. IETF RFC 3630. Septiembre 2003
- [5] R. Callon. *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*. IETF RFC 1195. Diciembre de 1990.
- [6] Satish Jamadagni. *OSPF extensions for flexible CSPF algorithm support*. Internet Draft. Octubre de 2002.
- [7] E. Rosen, A. Viswanathan, R. Callon. *Multiprotocol Label Switching Architecture*. IETF RFC 3031. Enero de 2001.
- [8] Bruce Davie, Yakov Rekhter. *MPLS. Technology and applications*. Morgan Kaufmann. 2000.
- [9] José Barberá. *MPLS: Una arquitectura de backbone para la Internet del siglo XXI*. Boletín RedIRIS. Nº 53. Septiembre de 2000.
- [10] Eric C. Rosen. *Exterior Gateway Protocol*. IETF RFC 827. Octubre de 1982.
- [11] D. L. Mills. *Exterior Gateway Protocol Formal Specification*. IETF RFC 904. Abril de 1984.
- [12] K. Lougheed, Y. Rekhter. *A Border Gateway Protocol (BGP)*. IETF RFC 1105. Junio de 1989.
- [13] Y. Rekhter, T. Li, S. Hares. *A Border Gateway Protocol version 4 (BGP4)*. IETF RFC 4271. Enero de 2006.
- [14] K. Varadhan. *BGP OSPF Interaction*. IETF RFC 1403. Enero de 1993.
- [15] T. Bates, R. Chandra, D. Katz, Y. Rekhter. *Multiprotocol Extensions for BGP-4*. IETF RFC 2283. Febrero de 1998.
- [16] Grupo GÍTACA. *Curso de Perfeccionamiento: "Administración Avanzada de Redes TCP/IP"*. Julio de 2007.
- [17] William Stallings. *High-speed networks. TCP/IP and ATM design principles*. Prentice Hall. 1998.
- [18] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow. *RSVP-TE: Extensions to RSVP for LSP Tunnels*. IETF RFC 3209. Diciembre de 2001.
- [19] A. Farrel, J.-P. Vasseur, J. Ash. *A Path Computation Element (PCE)-based Architecture*. IETF RFC 4655. Agosto de 2006.
- [20] OPNET Technologies Inc. (<http://www.opnet.com>)
- [21] OPNET Support Web Site. *Discrete Event Simulation Model Library*. (http://www.opnet.com/support/des_model_library/index.html)
- [22] The TOTEM Project Website. (<http://totem.info.ucl.ac.be/index.html>)
- [23] TOTEM Toolbox. *TOolbox for Traffic Engineering Methods*. (<http://totem.run.montefiore.ulg.ac.be>)
- [24] Y. Ikejiri, R. Zhang. *OSPF Protocol Extensions for Path Computation Element (PCE) Discovery*. IETF RFC 5088. Enero 2008
- [25] K. Kumaki, T. Murai. *BGP protocol extensions for Path Computation Element (PCE) Discovery in a BGP/MPLS IP-VPN*. IETF Draft. Abril 2008