UNIVERSIDAD CARLOS III DE MADRID

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

TESIS DOCTORAL

Mecanismos de protección en escenarios IP-MPLS multidominio

Autor: **Ricardo Romeral Ortega**

Director: **Dr. David Larrabeiti López**

Leganés, Julio de 2007

Mecanismos de protección en escenarios IP-MPLS multidominio

Autor: Ricardo Romeral Ortega

Director: Prof. Dr. David Larrabeiti López

Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad Carlos III de Madrid, el día __ de _____ de _____.

Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, __ de _____ de _____.

Índice de contenidos

Índice de figuras

Índice de tablas

Resumen

Una de las aplicaciones de ingeniería de tráfico más utilizadas en redes IP-MPLS es la protección de enlaces y nodos de la red mediante LSPs de modo que pueda llevarse a cabo una recuperación rápida del tráfico en casos de fallo. Las ventajas de utilizar MPLS en estos esquemas de protección frente a utilizar el reencaminamiento IP es bien conocida y documentada en la comunidad científica, fundamentalmente, rapidez de reacción, lo que conlleva una menor pérdida de tráfico.

Uno de los requisitos para poder proteger recursos en una red es ser capaz de encontrar un camino alternativo (o disjunto) al principal que no utilice los recursos protegidos. En los últimos años ha habido un esfuerzo por llevar estos esquemas de protección al interdominio. El principal problema para aplicar los esquemas intradominio a la protección de tráfico interdominio es el desconocimiento topológico de la red que impide obtener de manera sencilla los caminos de protección, disjuntos a los caminos principales. Parte de este desconocimiento viene dado por el modelo de encaminamiento interdominio por agregación utilizado en Internet basado en el protocolo BGP-4.

Existen en la literatura algunos esquemas distribuidos propuestos para solucionar este problema, pero todos ellos requieren compartir información topológica entre los dominios y la posibilidad de caer en topologías trampa, que sumen a estos mecanismos en un "prueba y error" retardando el establecimiento de los caminos. En la actualidad existe otra opción que está adquiriendo fuerza, la utilización de PCEs, entidades especializadas en computar caminos. La utilización de estos PCEs permite utilizar esquemas centralizados/distribuidos para la obtención de caminos disjuntos que sean utilizables en la protección de flujos de datos interdominio. Si bien no es una opción viable actualmente puesto que se está definiendo su arquitectura y protocolos básicos en el IETF en estos momentos.

En la presente Tesis Doctoral se proponen diversas soluciones que permiten obtener y señalizar dos caminos MPLS disjuntos (principal y de protección) interdominio. Se propone una modificación al protocolo BGP-4 de forma que a los dominios les sea posible tener información de dos AS_PATHs disjuntos que lleven a otro dominio. Si cada uno de los caminos sigue un AS_PATH distinto los caminos serán intrínsecamente disjuntos. Utilizando el modelo de encaminamiento actual se propo-

nen, además, dos esquemas distribuidos de cómputo y señalización de los caminos principal y de respaldo. El primero de ellos, evita las topologías trampa, tiene en cuenta la posible división en áreas de los diferentes dominios y respeta la privacidad entre los dominios, siendo un esquema especialmente rápido en el establecimiento de los caminos. El segundo de ellos, utiliza la protección interna de cada dominio para facilitar el cómputo de los caminos y reutilizar LSPs de respaldo. Finalmente, se realiza un estudio sobre cómo podrían utilizarse los p-ciclos propuestos por Stamatelakis y Grover para proteger los enlaces interdominio.

“Es de bien nacidos el ser agradecidos.”

Refranero popular

Agradecimientos

Mucha es la gente a la que hay que agradecer. Probablemente no a tantas personas directamente por la consecución en sí misma de esta Tesis pero sí a todas ellas de igual modo por la ayuda en la vida que termina con la consecución de la misma, puesto que en la vida, en esta parte de la mía, incluso en estos últimos meses, hay más cosas que la Tesis. De hecho, uno es capaz de terminarla porque las complementa con otras actividades. Es gracias a estas actividades de "escape" que luego es posible hacer todo lo demás, no sólo la Tesis. Y esas actividades de escape se hacen con amigos.

Mucha es la gente que me ha recordado de diversas maneras que había que terminar, que cuanto antes mejor. Son esas personas que cuando te preguntan por el tema les odias durante un instante infinitesimal de tiempo, se merecen ser los primeros.

La persona que más tiempo lleva insistiendo es, por supuesto, mi madre, Julia. A la zaga queda el resto de la familia, mi padre, Antonio, mis dos hermanos, Antonio y Eduardo. A todos ellos agradezco pertenecer a una familia genial, con los que se que siempre pude, he podido, puedo y podré contar. No me olvido de mis dos sobrinas, Ana y Ángela, dos soletes, y de su madre Esther.

La persona que más ha insistido en los últimos meses es, por supuesto, mi tutor, David. A la zaga quedan todos los compañeros, y en muchos casos amigos del Departamento; Manolo, Iván, Isaac, Isaías, Carlos Jesús, Antonio, Pablo, Alberto García, Paco, Jaime, Alberto Cortés, Ángel y Ruben, Carlos García, ...

La persona que más me ha insistido por unidad de tiempo que nos hemos visto, sin duda, ha sido Celia. Muchas gracias por tu amistad y apoyo continuo. A la zaga y recuperando terreno en los últimos meses esta Charly, uno de mis mejores amigos y personas que uno podría conocer. Sin duda, Cristina es la tercera en discordia, muchas gracias por todo.

La persona con la que la insistencia ha sido mutua, e incansable, en los últimos meses, ha sido Gorka, gran amigo, con quien he vivido infinitas anécdotas juntos.

Aunque ya han sido mencionadas quisiera agradecer especialmente a las personas que forman parte de mi grupo de trabajo, David, Manolo, Isaac y Ángel, por

haber soportado parte de mis responsabilidades estos últimos meses.

En mi estancia el año pasado conté con un anfitrión inigualable, gracias Dimitri.

Como decía al principio, “escapar” es fundamental, los grandes artífices de esto todavía no mencionados responden a los nombres de Héctor, Jorge, Jaime, Ramón, Félix, Alfredo, Hugo, Javi, Dani, …

Por último, pero no por ello menos importante, mi querida Ro, con quien he convivido y vivido este último año muchas experiencias. Todo lo que diga bueno sobre ella se queda corto, ella bien lo sabe.

Es posible que alguien quede en el tintero, en ese caso, ruego me disculpe y me lo comunique a la mayor brevedad posible, que la cuenta quedará saldada de inmediato con una buena conversación, aliñada de buena compañía y regada con rico “pan líquido”.

"La investigación se asemeja a los largos meses de gestación,
y la solución del problema, al día del nacimiento.
Investigar un problema es resolverlo."

Mao Tse Tung.
Contra el culto a los libros
(mayo de 1930)

Parte I

Planteamiento del problema

Capítulo 1

Introducción

Desde la aparición de las redes de comunicaciones una de las principales preocupaciones siempre ha sido cómo solucionar un fallo físico en algún equipo o enlace. Las primeras redes de comunicación utilizadas, los circuitos punto a punto, adolecían de un gran problema, el fallo de un enlace dejaba sin conectividad todas las comunicaciones establecidas por dicho enlace. El mismo problema tenían las redes de circuitos conmutados, si bien, en éstas existen caminos alternativos, las comunicaciones ya establecidas se perdían ante el fallo de un enlace.

La solución inicial a este problema pasa por la utilización de líneas de protección. Gracias a estas líneas podemos proteger enlaces y nodos que fallen y así poder seguir estableciendo comunicaciones. En las redes de circuitos conmutados las conexiones establecidas se perderán igualmente, sólo salvándose las nuevas conexiones. Para proteger también las conexiones establecidas se han utilizado dos formas distintas de actuar. Primera, mecanismos de protección más sofisticados que las líneas de respaldo, y segunda, la modificación de el concepto de red, de modo que sea ésta la encargada de ofrecer caminos alternativos a los intercambios de datos que se estén llevando a cabo.

La idea general detrás de los mecanismos de protección es la planificación de caminos alternativos que entran en funcionamiento en el momento del fallo. Esto es un paso más allá de disponer simplemente de una línea de respaldo. Existen diversos mecanismos de protección de enlaces diseñados para las redes de circuitos [1][2].

Una de las principales diferencias de las redes de conmutacion de paquetes respecto de las redes de conmutación de circuitos es que en éstas el camino que va a seguir la información se decide al inicio de la comunicación y toda la información enviada desde el emisor hasta el receptor sigue el camino estipulado. Es por este motivo que la red no ofrece por sí misma un mecanismo de recuperación de las comunicaciones en marcha, los mecanismos de recuperación de las comunicaciones deben ser extrínsecas a la red, diseñados a propósito. Sin embargo, en las redes de

conmutación de paquetes la información se divide en fragmentos, paquetes, que se envían de forma independiente, de modo que ante la caída de un enlace la red busca un nuevo camino para llegar al destino y los nuevos paquetes pueden llegar a este por el nuevo camino sin que se pierda la comunicación, aunque puede que sí parte de la información.

Al contrario de lo que pudiera parecer, la idea de que las redes de conmutación de paquetes son la panacea contra la pérdida de comunicaciones ante la caída de un enlace no es del todo cierta. El problema existente detrás es la respuesta a la siguiente pregunta: ¿cuánto tiempo tardará la red en encontrar el camino alternativo a utilizar? (en el caso de que exista). La respuesta ideal es que fuese cero. Pero, dado que esto es imposible la respuesta ideal realista es que fuese menor que el intervalo de llegada de paquetes al nodo, de modo que una vez caido el enlace, antes de que llegue el siguiente paquete, el nodo conozca el camino alternativo. Cuanto más cercano sea el tiempo final a este tanto mejor. Como haremos notar después, este tiempo de recuperación es, en muchos casos, muy superior al deseado, e incluso al permitido.

En las redes de paquetes existen protocolos y algoritmos de encaminamiento auxiliares al protocolo de red que les sirven a los nodos o routers para conocer los caminos que permiten llevar la información desde un origen a un destino, i.e. RIP (*Routing Information Protocol*) [3], OSPF (*Open Shortest Path First*) [4] o IS-IS (*Intermediate System to Intermediate System*) [5]. Estos protocolos en su mayoría son dinámicos, esto es, permiten adaptarse a cambios en la red. De este modo ante la caida de un nodo o un enlace los protocolos y algoritmos de encaminamiento, se ponen en marcha para obtener las nuevas alternativas que eviten el fallo. En este sentido estas redes son autónomas ante fallos, se adaptan a los cambios por sí mismas. El problema viene de la mano de el tiempo de reacción. Desde que ocurre el fallo en la red hasta que se tienen los caminos alternativos pueden pasar segundos, e incluso, en algunos casos, minutos [6].

Con la proliferación de Internet[1] la utilización de redes de paquetes se ha extendido ampliamente, no por sus cualidades como red respecto a la recuperación rápida de fallos, pero sí por sus cualidades como red con límite de usuarios blando, utilización eficiente de recursos, facilidad de ampliación, facilidad de aumentar la fiabilidad, etc, y también por su capacidad de recuperarse automáticamente y autonomamente ante fallos, aunque no sea de un modo especialmente rápido.

Precisamente, debido a la proliferación de este tipo de redes y la creciente demanda de servicios que se les pide a las redes de comunicaciones, cada vez se hace

[1]Durante esta memoria distinguiremos entre red de paquetes, red IP e Internet. Ningúna de las tres cosas es lo mismo, Internet es una red IP, pero no todas las redes IP son parte de Internet. Las redes IP son redes de paquetes, pero no todas las redes de paquetes son redes IP. Utilizaremos una u otra denominación según queramos enfatizar unas u otras cualidades.

más importante poder ofrecer desde la red un servicio de recuperación rápida ante fallos, no siendo suficiente el mecanismo simple de la propia red.

Como sabemos, Internet es una red de redes; esto, a grandes rasgos, quiere decir que distintas tecnologías de redes se unen para formar una única red. A cada una de estas redes, en terminología Internet, y en primera instancia, se las denomina dominios[2].

Dentro de estos dominios, los administradores, instalan nuevas tecnologías de red para proveer un mayor número de servicios y obtener una utilización más eficiente de la red. Una de estas nuevas tecnologías es MPLS (*Multiprotocol Label Switching*) [7], protocolo de conmutación de etiquetas que permite la creación de caminos virtuales por los que se enviarán los paquetes IP. Estos caminos pueden no ser los resultantes de ejecutar un protocolo de encaminamiento. De hecho, este es uno de los motivos por los cuales MPLS está adquiriendo mucha fuerza entre los operadores, dada su facilidad para ser utilizado como herramienta de ingeniería de tráfico sobre la red.

Actualmente, hay multitud de sistemas de recuperación de fallos dentro de un AS (*Autonomous System*) o Sistema Autónomo o región administrativa, que utiliza MPLS en su red [8]. Estos sistemas protegen un camino MPLS que se está utilizando para transportar flujos de un router a otro. Como veremos en la sección siguiente, hay sistemas más o menos escalables, sencillos o automáticos. La oferta es amplia y cada operador puede escoger el que más le convenga.

Dentro de un mismo dominio el reto a superar no es cómo conseguir que el camino principal (*primary LSP*) y el camino de respaldo (*backup LSP*) no compartan recursos entre sí (enlaces o nodos), es decir, que sean disjuntos. El algoritmo que obtiene el par de caminos disjuntos óptimo a partir del grafo completo de la red fue propuesto por Suurballe en el año 74 [9]. Los estudios actuales en el intradominio buscan cómo optimizar los caminos establecidos, minimizar la utilización de ancho de banda, minimizar el tiempo de respuesta o encontrar un compromiso entre estos factores.

Sin embargo, el problema de obtener estos caminos disjuntos es obtenerlos para proteger un flujo de datos que atraviese dos o más dominios [10][11]. Normalmente, la información topológica de un dominio no está disponible en el resto de dominios y no es factible utilizar directamente el algoritmo propuesto por Suurballe.

Normalmente, el cálculo de estos caminos disjuntos requiere de mecanismos distribuidos (i.e. PPRO (*Primary Path Route Object*) [12]) o centralizados/distribuidos (i.e. PCE (*Path Computation Element*) [13]), pero siempre existe una componente distribuida. Esta hace que el procedimiento de cálculo y señalización de los cami-

[2]Dominio se refiere a un conjunto de elementos de red que forman parte del mismo contexto administrativo, como por ejemplo, un Sistema Autónomo o un area IGP

nos se haga en colaboración, siendo las soluciones propuestas en el interdominio procedimientos de cálculo de caminos disjuntos muy dependientes de las fases de señalización de estos.

En el IETF (*Internet Engineering Task Force*) se están discutiendo algunas propuestas sobre cómo proveer sistemas de respaldo utilizando MPLS interdominio. En general, el interés mostrado por la comunidad es debido a que GMPLS (*Generalised MPLS*) será el protocolo que se utilizará en el futuro en las redes ópticas.

Hagamos un breve repaso a los grupos de trabajo del IETF relacionados con el desarrollo de MPLS o algunas de sus facetas.

Multiprotocol Label Switching WG (mpls) Este grupo de trabajo, que cuenta con más de 45 RFCs, se ha encargado de la estandarización de la tecnología de conmutación de etiquetas, así como de la implementación de la creación de LSPs (*Label Switched Path*) sobre las distintas tecnologías de nivel 2 (Frame Relay, ATM, Ethernet, etc), incluyendo el encapsulamiento y los procedimientos y protocolos de distribución de etiquetas. Actualmente, todos los elementos fundamentales de la arquitectura de MPLS (reenvío, cabeceras, señalización,...) están estandarizadas, y las preocupaciones actuales del grupo son, entre otras, las extensiones para MPLS P2MP (*Point to Multipoint*), la propuesta de mecanismos para MPLS OAM (*Operations and Management*) o estudiar, conjuntamente con el grupo CCAMP, los aspectos de ingeniería de tráfico para multiples áreas y multiples dominios. [14].

Common Control and Measurement Plane WG (ccamp) Con alrededor de 25 RFCs estandarizadas, es el grupo encargado de coordinar dentro del IETF los trabajos de definición del plano de control común de caminos físicos y túneles de ingeniería de tráfico utilizados por los proveedores de servicios. Este plano de control común incluye la adquisición y distribución de los atributos necesarios para establecer los caminos y túneles. Entre las tareas más relevantes de coordinación de este grupo están; ampliar los protocolos de encaminamiento (OSPF, IS-IS) y señalización (RSVP-TE) necesarios para soportar el cálculo y establecimiento de caminos; definir, junto a los grupos de trabajo MPLS y PCE la señalización y los mecanismos de routing para poder establecer caminos y túneles interdominio; y estudiar la migración de MPLS a GMPLS [15].

Path Computation Element WG (pce) Creado en Agosto de 2004, es el grupo de trabajo encargado de definir una arquitectura de cálculo de caminos para hacer ingeniería de tráfico usando LSPs GMPLS. Además de la arquitectura, el grupo se encarga de definir los protocolos de comunicación entre los distintos PCEs o entre un PCE y un PCC (*Path Computation Client*). La arquitectura definida esta

pensada para un sólo dominio o para la comunicación entre un conjunto reducido de dominios [16].

Layer 2 Virtual Private Networks WG (l2vpn) Dentro de este grupo se intentan definir y especificar diversas soluciones para proveer redes privadas virtuales de nivel 2 (L2VPNs). En particular el grupo es responsable de estandarizar una solución para proveer un servicio de LAN privada virtual, VPLS (*Virtual Private LAN Service*), sobre capa 2 que emule una LAN sobre una red IP o una red MPLS. Una de las restricciones más importantes del grupo es no estandarizar ningún nuevo protocolo. El WG debe obtener una solución válida sólo extendiendo protocolos ya estadarizados [17].

Layer 3 Virtual Private Networks WG (l3vpn) La función de este grupo de trabajo es la estandarización de varias soluciones de red privada virtual de capa 3. Una de estas soluciones debe proveer una VPN (*Virtual Private Network*) IP en una red BGP/MPLS. Dentro de los escenarios a estudiar por este grupo de trabajo se incluyen varios en los que diferentes dominios deben colaborar [18].

Inter-Domain Routing WG (idr) y Secure Inter-Domain Routing WG (sdir) Son los grupos de trabajo encargados de la estandarización de BGP-4 y de los aspectos de seguridad en el routing interdominio respectivamente [19] [20]. Este último debe definir una arquitectura de encaminamiento segura, tanto para unicast como para multicast, teniendo en cuenta la practicidad de la solución, es decir, debe poder ser implementable. En cuanto al primero de los grupos de trabajo, IDR, mantiene la actualización, promoción y usabilidad de BGP-4 como principales objetivos. El segundo, SIDR, se encarga de los aspectos de seguridad.

A continuación pasaremos a explicar someramente cómo funciona MPLS y algunas de sus aplicaciones más importantes. Primero, explicaremos las soluciones de protección de intradominio que más se utilizan actualmente, para pasar a plantear las soluciones interdominio que se están discutiendo en el IETF. Terminaremos exponiendo el camino que seguirán nuestras investigaciones entorno al trabajo de Tesis Doctoral a que este documento se refiere.

Capítulo 2

Estado del arte

2.1. MPLS y sus aplicaciones

En este apartado explicaremos brevemente cómo funciona MPLS y algunas de sus principales aplicaciones. Con esto trataremos de ofrecer una visión de las posibilidades que ofrece MPLS como herramienta de ingeniería de tráfico. Una de las principales características de MPLS es aumentar la disponibilidad de la red. Esta disponibilidad puede verse aumentada si, además, interactuan los distintos dominios entre sí realizando ingeniería de tráfico interdominio.

2.1.1. Funcionamiento general de MPLS

MPLS es un protocolo de comunicaciones habitualmente situado entre los niveles de enlace y red basado en la conmutación de paquetes mediante etiquetas.

Una red MPLS consiste en un conjunto de LSRs (*Label Switching Router*) conectados entre sí. Cuando un paquete entra en la red, el LSR frontera obtiene la FEC (*Forwarding Equivalency Class*) del paquete a partir de la información contenida en las cabeceras de nivel superior. A partir de la FEC el LSR frontera asigna una etiqueta al paquete y obtiene el siguiente salto. De este modo el LSR de borde ha introducido el paquete (lo mismo hará con todos aquellos con la misma FEC) en un LSP que llegará a un router de salida del dominio MPLS. Esto es debido a que cada router MPLS del camino toma la decisión de reenvío basándose únicamente en la etiqueta del paquete, con lo que todos los paquetes con igual FEC y con el mismo LSR de entrada a la red siguen el mismo LSP o camino MPLS [21].

MPLS, al igual que ATM (*Asynchronous Transfer Mode*) o Frame Relay, es una tecnología de conmutación de paquetes basada en circuitos virtuales. Más adelante veremos las aplicaciones más importantes de MPLS, pero muchas de ellas simple-

mente son derivadas de las posibilidades de trabajar con túneles no dependientes de IP y poder obviar la información de reenvio del nivel de red. A diferencia de ATM o Frame Relay, MPLS no depende de ninguna tecnología de nivel inferior o superior concreta. Debido a esta naturaleza multiprotocolo su implantación es más sencilla, ya que no es necesario, entre otros motivos, modificar el resto de tecnologías asociadas.

La principal ventaja de utilizar MPLS que se suele argüir inicialmente es que la conmutación de etiquetas es más rápida que las búsquedas en las tablas de encaminamiento IP [22]. Esto era cierto hace algunos años, pero actualmente, la utilización de harware específico, como las TCAMs (*Ternary Content Addressable Memory*) [23], hace que el tiempo consumido en las búsquedas en las tablas de forwarding IP sea similar al tiempo de búsqueda en las tablas de etiquetas MPLS [24].

Hay investigaciones que van incluso más allá, en este trabajo reciente [24] Persaud et al. muestran cómo incluso utilizando un *Network Processor* IXP1200 [25] sin hardware optimizado para las búsquedas IP (CAM o TCAM) hay casos para los que el retardo en el reenvio en un LSR es similar al de un router IP. Las pruebas de Persaud muestran que dependiendo de la velocidad de la línea de salida y el tamaño del paquete IP, en general, para paquetes IP mayores de 150 octetos, el reenvío IP y el reenvío MPLS sufren retardos similares.

Pero las utilidades de MPLS van más allá que la simple velocidad en la conmutación de etiquetas. En el *MPLS-VPLS Resource Center* [8], disponen de una lista de ventajas/aplicaciones consideradas como las más importantes para utilizar MPLS. De estas, a continuación, vamos a ver alguna de las más importantes.

2.1.2. Aplicaciones

Voz sobre MPLS. Hay dos vías para llevar la voz sobre una red MPLS (VoMPLS). La primera de ellas, VoIP (*Voice over IP*) sobre MPLS, es encapsular las tramas de VoIP sobre una red MPLS. En este caso la red MPLS simplemente realiza el transporte. En algunas implementaciones es posible realizar compresión de cabeceras para optimizar el envío [26].

La segunda vía, VoMPLS (*Voice over MPLS*) propiamente dicho, es encapsular las muestras de voz directamente sobre MPLS. En [27] se explica cómo realizar esta operación, así como, cómo encapsular la señalización, el canal asociado o los dígitos de marcación DTMF (*Dual Tone Multi-Frequency*).

VPLSs. Una VPLS es un servicio Ethernet global, nacional o regional utilizado para extender la red de área local de un cliente a multiples sitios remotos ofreciendo una conectividad multipunto completa [28]. Es la RFC 4448 [29] la que explica

cómo encapsular Ethernet para transportarlo en redes MPLS.

Desde el punto de vista de los proveedores de servicio, implementar el servicio de VPLS sobre MPLS lo convierte en una solución al problema eficiente, escalable y segura. Los clientes pueden beneficiarse de ventajas adicionales, como poder obtener una solución de conectividad multipunto económica con posibilidad de ofrecer calidad de servicio [30].

En [31] y en [32] se definen los mecanimos necesarios para el autodescubrimiento de PEs (*Provider Edge Router*) y la creación de los *pseudowires*, conexiones punto a punto entre los PEs. Para la señalización de VPLSs se utiliza BGP [31], definiéndose un BGP NLRI (*Network Layer Reachability Information*). Para la información de VPLSs se utiliza LDP (*Label Distribution Protocol*) [32], en el que se define un nuevo TLV (*Type Length Value*).

Uno de los temas objeto de atención por los investigadores sobre una red VPLS es el servicio multicast. [33] [34] proponen un algoritmo para el cálculo de árboles multicast en un dominio VPLS que optimice el número de estados multicast en los routers de la red VPLS de modo que sea más escalable que otras soluciones tradicionales. Esta optimización, a cambio, recae en un aumento del uso del ancho de banda, puesto que se basa en el cálculo de un árbol agregado para varias VPLSs cuyo objetivo de construcción es obtener el árbol de menor retardo.

Los túneles que propone utilizar el IETF para la red VPLS son punto a punto. Algunos investigadores, en cambio, proponen la utilización de túneles punto a multipunto, para optimizar la utilización de la red, proponiendo algoritmos para minimizar la utilización del número de caminos necesarios [35] [36] [37].

La creación y establecimiento de tuneles punto a multipunto deben proveerlos los protocolos de señalización. En el grupo de MPLS del IETF se esta definiendo una extensión para RSVP-TE (*RSVP Traffic Engineering*) que soporte este tipo de túneles [38]. Gracias a la posibilidad de crear estos túneles aparece un nuevo espacio de soluciones posibles para implementar VPLSs.

Moerman et al. [39] han mostrado, gracias a la red UTOPIA, que es posible utilizar una red VPLS basada en MPLS para ofrecer servicios "*Triple Play*".

VPNs. Como muestra de la importancia que están adquiriendo recientemente las VPNs en MPLS veremos la evolución de las RFCs con dos actualizaciones en 2006. En la RFC 2547 [40], de marzo de 1999, se definió como un operador podía crear VPNs en su *backbone* IP utilizando MPLS como tecnología de reenvío y BGP como protocolo de intercambio de rutas. Posteriormente esta RFC fue reemplazada por la RFC 4364 [41], de febrero de 2006, y esta ha sido recientemente sustituida, en junio de 2006, por la RFC 4577 [42]. Esta RFC tiene como principal cambio ofrecer la posibilidad de utilizar OSPF como protocolo de encaminamiento entre los routers

del cliente, CEs (*Customer Edge Router*), y los routers del proveedor, PEs, de modo que el cliente no necesite configurar BGP para poder utilizar la VPN.

En muchos de los temas que se estudian de forma genérica para MPLS algunos investigadores buscan su aplicación concreta a las VPNs. Así, por ejemplo, Siriakkarap et al. en [43] proponen una técnica de protección de caminos para VPN basada en dos nuevos objetos RSVP (*Resource Reservation Protocol*), el objeto DETOUR y el objeto CONSTRAINT, con los que obtienen el camino de respaldo óptimo con menor carga media de CPU de los routers utilizados sin incurrir en un mayor número de saltos.

En Cisco han llevado a cabo estudios sobre la seguridad de estas redes. En [44] se hace un estudio comparativo con VPNs basadas en ATM o Frame Relay, llegando a la conclusión de que a pesar de que el control en MPLS esté basado en la capa 3, es posible ofrecer el mismo tipo de seguridad que en aquellas.

En 2004, Ren et al. [45] analizaron los problemas de seguridad de las VPNs basadas en MPLS y propusieron un esquema de securización basado en IPSec (*Internet Protocol Security*) que no tiene casi contraprestaciones en la calidad de la red, retardo, tráfico enviado y sobrecarga de cabeceras.

No sólo se busca la seguridad global de la red, sino que también se busca la integración de redes seguras con redes no seguras, así, por ejemplo, en [46] se explica cómo transportar de manera segura tramas MPLS utilizando IPSec en el núcleo de la red. Aunque luego el transporte pueda ser MPLS dentro de la red, con lo que tendríamos, una trama MPLS con un paquete IP en su interior que contiene un paquete IPSec con la trama MPLS de la VPN en su interior.

QoS. Existen distintos esfuerzos por integrar MPLS en redes con QoS, tanto DiffServ como IntServ. Rouhana y Horlait fueron de los primeros en proponerlo [47].

En [48] se describe cómo dar soporte a DiffServ en una red MPLS. Nótese que no es una tarea trivial puesto que en DiffServ la etiqueta que determina el PHB (*Per Hop Behaviour*), esto es, el comportamiento que debe tener el router frente a un paquete, se encuentra en la cabecera IP, cabecera que no se inspecciona en un router MPLS *interior*. Esto lleva al otro problema importante: cómo mapear el campo DSCP de DiffServ/IP (6 bits) en el campo EXP de MPLS (3 bits). En [48] están estandarizados dos modos de realizar esta tarea, E-LSP (*EXP-Inferred-PSC LSP*), y L-LSP (*Label-Only-Inferred-PSC LSP*).

Avallone et al. han realizado una comparación en [49] de las prestaciones sobre una red de pruebas de cinco PCs en diferentes escenarios; *Best Effort*, DiffServ, MPLS y DiffServ en MPLS, dando resultados esperanzadores para el último de estos.

Un aspecto importante en toda arquitectura de calidad de servicio es cómo implementarla en el router. En [50] se describe el diseño e implementación de los principales componentes para proveer calidad de servicio, i.e. planificador, clasificador, acondicionador, etc., de un router MPLS *frontera* basado en un *procesador de red*.

Fabricantes de *procesadores de paquetes* de propósito general, como Intel, están empezando a sacar al mercado *procesadores de red* con implementaciones de DiffServ sobre MPLS [51].

En el caso de ser necesario garantizar cierta calidad de servicio durante un fallo, se deben tener en cuenta ciertas consideraciones. Por ejemplo, en el caso de realizar reservas de caminos con restricciones de QoS el orden del cálculo de los caminos puede llevar a obtener una solución para todos ellos o a no encontrarla, a pesar de existir una. En [52], sección 5.15, se realiza un estudio detallado de estas consideraciones, explicando cúales son y cómo se deberían afrontar en una implementación real.

2.2. Sistemas de protección. Terminología

Reencaminamiento vs. protección. La recuperación basada en reencaminamiento calcula y establece el camino o segmento de respaldo bajo demanda después de ocurrir el fallo (aunque en ocasiones el cálculo puede ser anterior al fallo, no así la señalización del establecimiento, estaríamos distinguiendo entre caminos preplanificados y calculados dinámicamente [52]). En cambio, en una recuperación basada en protección, el camino o segmento de respaldo ha sido calculado y establecido previamente al fallo. Sólo en caso de fallo el tráfico se conmuta al camino preestablecido [53].

Recuperación global vs. recuperación local. En todo tipo de recuperación existen dos LSRs clave. El PSL (*Path Switch LSR*) que es el LSR encargado de conmutar o duplicar el tráfico que viene del *camino principal* al *camino de respaldo*[1]. Y el PML (*Path Merge LSR*) que es el LSR que restituye el tráfico que llega por el camino de respaldo al camino principal. En otras palabras, el camino de respaldo se situa entre el PSL y el PML sustituyendo el segmento de camino principal que estaba situado entre estos LSRs.

En una recuperación local sólo se busca alternativa a los elementos que han fallado, es decir, el PSL y el PML se situan lo más cerca posible al elemento de red que haya fallado. En caso de ser un enlace serán los LSRs adyacentes al enlace. En

[1] En el apéndice A se definen los conceptos de *camino principal* (definición A.10) y de *camino de respaldo*(definición A.11).

el caso de ser un nodo el que falle el PSL y PML serán los LSRs vecinos de dicho nodo que ha fallado.

En una recuperación global se sustituye todo el camino principal por el camino de respaldo. Esto significa que el PSL será el LSR de entrada del camino principal y el PML será el LSR de salida del camino principal [52][2].

Como es evidente, las recuperaciones local y global son los extremos de una serie de posibilidades, en las que se pueden recuperar segmentos de mayor o menor longitud.

Tipos de protección Podemos dividir los mecanismos de recuperación basados en protección en cinco subtipos. Cada uno de estos subtipos además puede referirse a dos conceptos diferentes, reserva de recursos o asignación de protecciones. Así, por ejemplo, en el tipo de protección 1:1 podemos referirnos a que los recursos reservados en el camino de respaldo están reservados al tráfico de un solo camino principal en caso de fallo o simplemente que un camino de respaldo sólo tiene asignado un camino principal a proteger. En las descripciones de los tipos de protección que realizaremos a continuación utilizaremos la terminología relativa a la asignación de protecciones, con la reserva de recursos sería igual.

1+1 Un camino de respaldo sólo protege un camino principal. Además, el tráfico del camino principal es enviado simultaneamente por el camino de respaldo. Es el PML el encargado de seleccionar en cada momento de cuál de los caminos obtener el tráfico.

1:1 Un camino de respaldo sólo protege un camino principal. En este caso sólo se envia el tráfico por el camino de respaldo si hay un fallo en el camino principal.

1:n Un camino de respaldo protege *n* caminos principales. Estos caminos principales pueden tener trayectos distintos siempre que tengan en común el PSL y el PML.

m:n *m* caminos de respaldo protegen *n* caminos principales. Estos caminos principales pueden tener trayectos distintos siempre que tengan en común el PSL y el PML.

Camino dividido Varios caminos de respaldo se utilizan para llevar el tráfico de un camino principal. Se puede configurar el ratio de tráfico a transportar por cada camino de respaldo. Este tipo de protección se utiliza cuando se quieren reservar recursos y no se pueden asegurar con un solo camino. El PML es el

[2] Aunque las ideas generales sobre recuperación global y local se hayan obtenido de [52] la nomenclatura utilizada es la propuesta en [53] puesto que es la más extensamente utilizada, incluidas las RFCs, en el ambito de MPLS.

encargado de reordenar el tráfico que llega desde los diferentes caminos de respaldo.

2.3. Sistemas de protección intradominio

En este apartado veremos algunas de las propuestas de protección intradominio, y que son la idea base para algunos de los mecanismos interdominio.

En la RFC 4090 [54] se definen dos esquemas de *fast reroute*, es decir, de protección local[3]. Utilizaremos la red MPLS de la figura 2.1 para exponer los ejemplos de ambos esquemas.

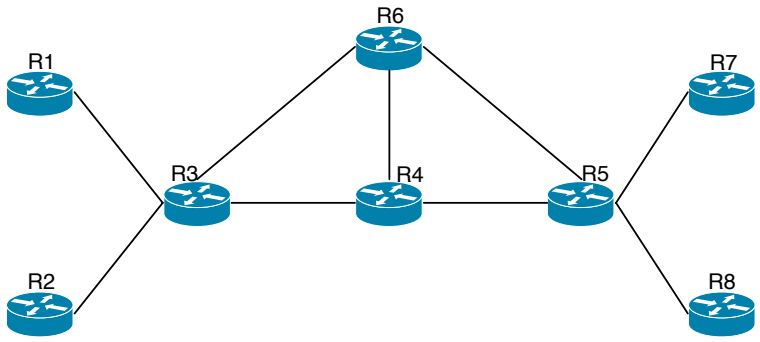


Figura 2.1: Red MPLS ejemplo

One-to-one Este esquema de recuperación reserva un camino de respaldo, denominado *desvío*, para cada camino principal, denominado *camino protegido*. Por tanto, estamos ante un esquema 1:1. Veamos cómo funciona con ayuda de un ejemplo. En la figura 2.2 podemos ver dos *caminos protegidos*, CP1 (R1-R3-R4-R5-R7) y CP2 (R2-R3-R4-R5-R8). Para proteger estos caminos ante fallos en el enlace R3-R4 o en el nodo R4 son necesarios 2 *desvíos*, D1 (R3-R6-R5-R7) y D2 (R3-R6-R5-R8).

Si se quiere proteger frente a fallos en todos los enlaces y nodos de un camino es necesario realizar los *desvíos* en todos los LSRs pertenecientes a un camino.

En el ejemplo propuesto, cuando existe un fallo en R4 el PSL R3 en lugar de realizar la conmutación de etiquetas para continuar por el camino CP1, conmuta las etiquetas de los paquetes de ese camino por las del camino D1 y lo envía por

[3]Aunque el nombre pueda llevar a error, *fast reroute* se refiere a un esquema de recuperación basado en protección. Tener los caminos de respaldo preestablecidos es la única manera de poder realizar recuperación "rápida".

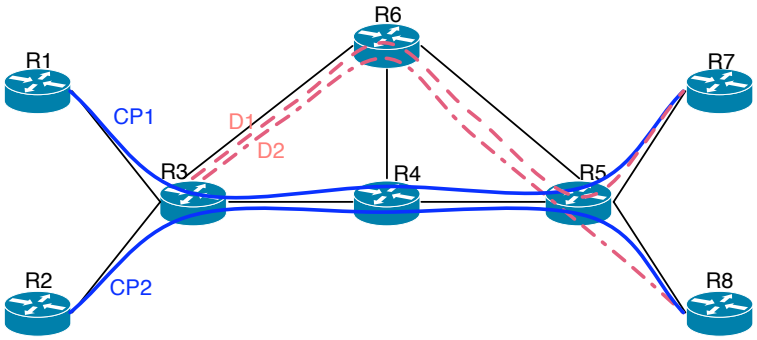


Figura 2.2: Ejemplo esquema *fast reroute one-to-one*

la interfaz R3-R6, de modo que los paquetes quedan conducidos por D1 hasta el destino final. Para CP2 y D2 se realiza lo mismo.

En la RFC se recomienda que siempre que sea posible los *desvíos* se encuentren con los *caminos protegidos* y realizar en el PML un *merging* de etiquetas. La finalidad de esto es reducir el numero de entradas de reenvío en los routers para los *desvíos*.

Facility backup Este esquema de recuperación permite compartir un camino de respaldo, denominado *bypass* para varios *caminos protegidos*. Por tanto, estamos ante un esquema 1:*n*. Igual que en el caso anterior nos ayudaremos de un ejemplo, mostrado en la figura 2.3, para ver su funcionamiento.

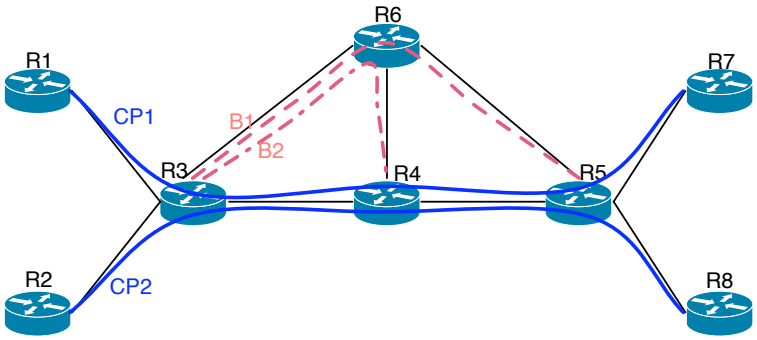


Figura 2.3: Ejemplo esquema *fast reroute facility backup*

Al igual que antes tenemos dos *caminos protegidos*, CP1 y CP2. En la figura podemos ver que existen dos caminos de *bypass*, B1 (R3-R6-R5) y B2 (R3-R4). Normalmente se dice que el camino B2 es de tipo NHOP (*Next Hop*) y el camino B1 de tipo NNHOP (*Next-Next Hop*). B2 sólo protege frente a caídas del enlace R3-R4, en cambio, B1, además, protege frente a fallos en el nodo R4.

El funcionamiento de ambos *bypass* se basa en la utilización de *label stacking* de MPLS. En el caso de un *bypass* NHOP, como es B2, el PML se situa en el nodo adyacente al enlace protegido. Cuando el enlace falla y es necesario activar el mecanismo de recuperación , el nodo PSL, en el ejemplo R3, conmuta la etiqueta del paquete como hace normalmente y luego añade la etiqueta del tunel de *bypass* enviandolo a R6. Cuando R6 recibe el paquete reenvía este a R4 haciendo cambio de etiquetas o utilizando PHP (*Penultimate Hop Popping*). Este paquete una vez en R4 y fuera del tunel de *bypass* tiene la misma etiqueta que esperaría R4 si hubiese llegado por CP1 o CP2, aunque ha llegado por otra interfaz, pero con asignación de etiquetas *per platform* es indiferente de dónde haya llegado el paquete.

En el caso de ser un *bypass* NNHOP, como es B1, el PML se situa en el nodo siguiente al que se quiere proteger dentro de los caminos protegidos. Es por esto que, utilizando *bypass* NNHOP, los *caminos protegidos* deben compartir dos enlaces y un nodo. En el caso de *bypass* NHOP sólo es necesario que compartan un enlace. El funcionamiento de este tunel en caso de fallo es similar al anterior, R3 realiza un cambio de etiquetas y utilizando *label stacking* envia el paquete por B1 hasta R5. Una vez el paquete está en R5 y fuera del tunel de *bypass* el paquete sigue el curso normal. Nótese que para que esto funcione la etiqueta con la que debe recibir el paquete R5 y que conmuta R3 antes de introducir tráfico en el tunel es la que esperaría de R4, etiqueta que sólo conocen R4 y R5. Es por esto que se deben proveer mecanismos para que R3 pueda descubrir esta etiqueta y pueda utilizarla en caso de fallo, de este modo si R5 dispone de asignación de etiquetas *per platform*, al ser indiferente el interfaz de entrada de los paquetes, continuaría enviando el tráfico por el túnel original. En la RFC 4090 [54] se describen las extensiones necesarias a RSVP-TE para poder realizar este descubrimiento de etiquetas y poder establecer tanto protecciones *one-to-one* como *facility backup*.

Este esquema de protección es mucho más escalable que *one-to-one*, debido a que con un solo *bypass* podemos proteger todos los caminos que circulan por un enlace y/o nodo, con independencia de su número. Por supuesto, es posible elegir qué caminos se quieren proteger con un determinado *bypass*, no es necesario que sean todos los que comparten el enlace o segmento a proteger.

Otros tipos de recuperación [55] es un interesante artículo que realiza una panorámica de los mecanismos de protección más conocidos de MPLS. Aquí solamente introduciremos los conceptos de dos de ellos a modo de ejemplo.

Makam et al. [56] propusieron uno de los primeros métodos de protección para redes MPLS. Es un método de recuperación global básico. Éste puede utilizarse como método de protección y como método de reencaminamiento. El funcionamiento básico consiste en que el nodo que detecta el fallo envía un aviso al LSR de entrada del LSP protegido para que éste conmute el tráfico, y, en su caso lo establezca, por el

camino de respaldo. El principal problema de este esquema es que el nodo que detecta el fallo debe enviar un aviso al LSR de cabecera (*head-end*). Durante el tiempo que tarda esta notificación es posible que se pierdan muchos paquetes.

El esquema propuesto por Hasking et al. [57] intenta paliar el problema del esquema anterior. El nodo que detecta el fallo reenvía el tráfico a la cabecera del LSP, por el llamado *reverse backup*, y una vez ahí se reenvía por el camino de respaldo. La pérdida de paquetes con este sistema se minimiza, puesto que es el propio nodo que detecta el fallo quien realiza la operación de recuperación. Además no es necesario implementar un mecanismo de notificaciones de fallo entre LSRs. Por contra, el consumo de ancho de banda es mayor puesto que hay por parte del camino que el tráfico va y vuelve durante el fallo, desde la cabecera hasta el PSL y desde éste de nuevo hasta la cabecera para introducirlo en el camino de respaldo.

Existen muchos otros esquemas de recuperación ante fallos en redes MPLS, algunos de ellos proponen la creación de árboles de notificación [58] o la concatenacion de caminos para la protección de multiples fallos simultaneos [59].

P-ciclos La tecnología de p-ciclos fue introducida por W. D. Grover y D. Stamatelakis para la recuperación rápida de enlaces en el año 98 [60]. Consiste en la preconfiguración de anillos de recuperación cuya virtud es que su distribución por una red mallada es tal que son capaces de ser utilizados para recuperar cualquier enlace de la red. Este primer trabajo estaba planteado para la recuperación de los enlaces a nivel físico, intentaban obtener los tiempos y simplicidad de recuperación de las redes en anillo en una red mallada.

Los p-ciclos se presentaron como un mecanismo con la principal ventaja de los esquemas de recuperación de las redes de anillo, la velocidad de recuperación, y la principal ventaja de los esquemas de recuperación de redes malladas, el uso eficiente de la capacidad reservada para el respaldo.

Los p-ciclos son una tecnología de cómo proteger una red de tipo malla utilizando anillos de protección preconfigurados, por tanto, no es una tecnología exclusiva de MPLS, de hecho las primeras ideas fueron para utilizarlos en redes tipo WDM y Sonet [60]. Posteriormente fue propuesto como esquema para redes MPLS/IP [61].

Originalmente los p-ciclos se concibieron como protección frente a caídas de enlaces. Más tarde se ha estudiado también cómo utilizar los p-ciclos para protegerse de la caida de nodos utilizando los "*p-ciclos rodeando un nodo*". Este nuevo concepto se presentó por primera vez en [61]. Consiste en la creación de un p-ciclo utilizando los nodos que rodean completamente el nodo que se quiere proteger. Posteriores trabajos realizan estudios más avanzados, por ejemplo, de cómo máximizar la protección de enlaces y nodos utilizando sólo "*p-ciclos rodeando un nodo*" [62].

Además de extensiones a la protección de nodos también se han propuesto so-

luciones para la protección de segmentos de camino [63]. En este trabajo, Grover y Shen extienden el concepto de p-ciclo de modo que lo que se proteja no sean enlaces sino caminos, más acorde a las técnicas de recuperación de MPLS. Estos nuevos p-ciclos los denominan "*p-ciclos de flujo*", puesto que realmente si un camino lleva un flujo estos p-ciclos servirían para proteger flujos completos, y no enlaces, es un concepto de protección completamente diferente.

Uno de los temas más complejos de los p-ciclos es determinar cuántos y qué p-ciclos utilizar para proteger una red completa de la manera más eficiente. Existen diversos trabajos que van en esta línea [64] [65] [66].

Tal es esta búsqueda de eficiencia que se han llegado a definir los p-ciclos no-simples, de modo que con un menor número de p-ciclos o utilizando menos capacidad libre de la red se puede obtener un nivel similar de protección. También permiten proteger zonas de una red que no podrían protegerse mediante p-ciclos simples. Un p-ciclo no-simple consiste en un p-ciclo que puede pasar por un nodo y/o enlace varias veces [67].

2.4. Sistemas de protección interdominio

En el IETF se han presentado diferentes trabajos [68][69][70] dedicados al estudio de las particularidades del interdominio en las redes de nueva generación que incluyen MPLS y redes ópticas. [71] resume el trabajo del OIF (*Optical Internetworking Forum*) y del IETF en redes ópticas multidominio. En particular, este estudio se centra en las restricciones y consideraciones especiales de las redes ópticas transparentes y en los temas asociados con las redes ópticas multidominio. [68] describe brevemente los distintos tipos de fallos que deben ser resueltos por un mecanismo de *fast rerouting* interdominio.

Una de las primeras soluciones propuestas al problema de la protección de LSPs en el interdomino fue la de Huang et al. en [72]. La solución propuesta se basa en la utilización de mecanismos de protección independientes en el interior de los dominos y en sus fronteras. En el interior de los dominios se utilizaría un LSP principal y uno de respaldo. Para protegerse de fallos en el interdominio se utiliza un mecanismo convencional de recuperación local. Este mecanismo ha sido llamado por los autores IBLBT (*InterDomain Boundary Local Bypass Tunnel*). En la figura 2.4 se muestra un ejemplo de funcionamiento de este mecanismo. Sólo comentaremos que, como puede verse en la figura, para proteger un camino principal es necesario utilizar 4 caminos de respaldo, siendo necesario señalizar 3 de ellos en el interdominio.

Hay que destacar que este tipo de solución, no es una solución extremo a extremo, sino que se basa en recuperación local de enlaces. Los trabajos posteriores

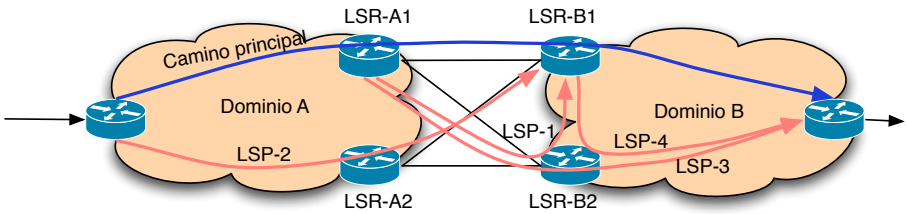


Figura 2.4: Ejemplo de recuperación rápida intra e inter dominio (IBLBT)

abogan, en su mayoría, por recuperación de tipo global para LSPs interdominio.

Farkas et al. están estudiando la posibilidad de utilizar p-ciclos en el interdominio [73]. La idea sería configurar p-ciclos que unan dominios, y utilizando algún tipo de protección independiente en el interior de los dominios. De forma que el p-ciclo interdominio entra en juego cuando se cae alguno de los enlaces o nodos interdominio.

Uno de los problemas detectados más importante cuando se trabaja con distintos dominios no es sólo proteger un LSP interdominio, sino ser capaz de obtener un LSP de respaldo que sea capaz de proteger un LSP principal; esto es, que sea disjunto en recursos de red (nodos y enlaces). Por este motivo, los trabajos más recientes en el interdominio también tienen en cuenta el mecanismo de cálculo del camino disjunto que servirá de protección.

Anteriormente, se han propuesto distintas soluciones para resolver el cálculo de un camino TE interdominio [68], que en sí mismo ya es complejo. También, como hemos visto, se han propuesto algunas soluciones a la protección interdominio. Pero, debido a la complejidad que requieren las soluciones al problema, los trabajos más recientes unen el cálculo de los LSPs con la señalización de los mismos y con el tipo de protección a utilizar, dando una solución completa. A continuación vamos a explicar algunas de las propuestas presentadas hasta ahora.

2.4.1. Método basado en PCE

En el año 2005 el IETF inició un nuevo grupo de trabajo llamado PCE WG [16]. El PCE es un elemento de red utilizado para calcular caminos MPLS o GMPLS dentro de un dominio. El camino a calcular puede ser solicitado por un nodo del dominio o por un nodo autorizado de otro dominio. De este modo los PCEs pueden ser utilizados para calcular y señalizar caminos MPLS interdominio extremo a extremo.

Si bien la arquitectura de PCE empieza a estar plenamente definida [13], no así sus posibles aplicaciones. Cómo utilizar la arquitectura PCE para obtener y señalizar caminos MPLS interdominio, de momento, no está estandarizado, todo lo que hay en la literatura son propuestas de uso particulares. Según propone Ricciato et al. en [10]

el procedimiento podría ser el que describimos a continuación. Nos ayudaremos de un ejemplo basado en la figura 2.5.

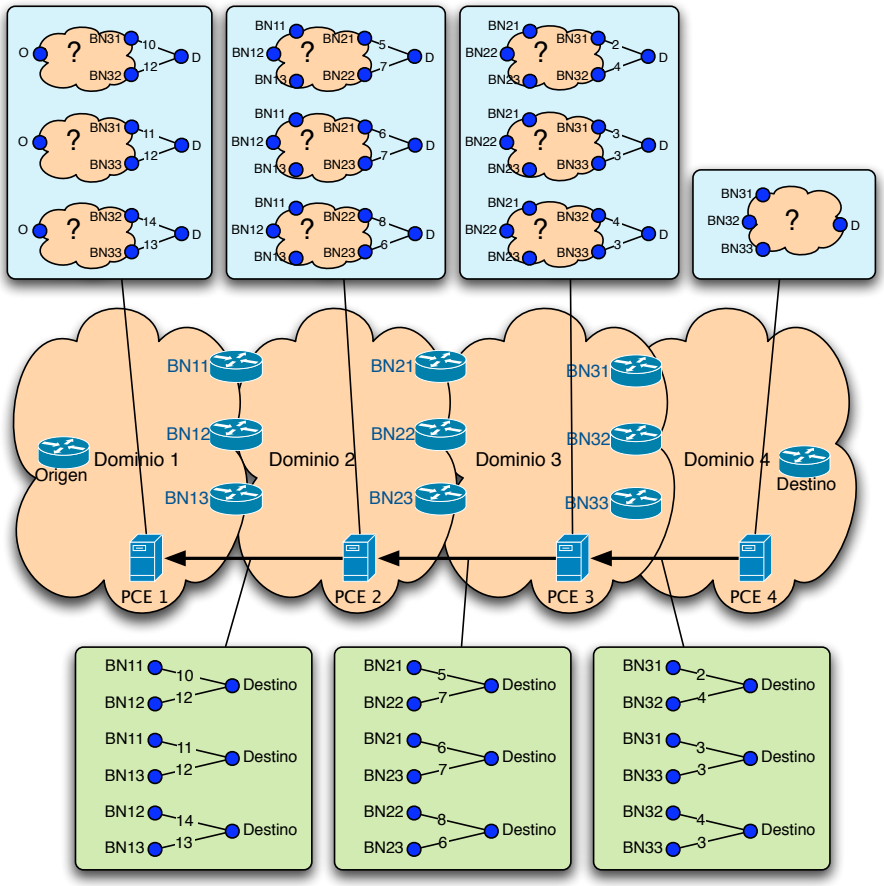


Figura 2.5: Ejemplo de utilización de PCE

El router cabecera solicita a su PCE un par de caminos disjuntos a un determinado destino a través de una secuencia de ASes, *AS Path*. Este PCE reenvía la petición al PCE del siguiente dominio. Así sucesivamente hasta que se alcanza el PCE del dominio destino, el *dominio 4*. Este último PCE calcula todos los posibles pares de caminos disjuntos dentro de su dominio y envía dicha información al PCE del que recibió la petición. Éste realiza el mismo cálculo teniendo en cuenta la información recibida, de modo que envía al PCE anterior todos los pares de caminos disjuntos posibles hasta el destino. Así sucesivamente hasta que el *PCE 1* obtiene el par de caminos disjuntos óptimo extremo a extremo que le envía al nodo que realizó la petición.

Este método se denomina “pensar antes de romper” porque el establecimiento del LSP se posterga hasta que se tiene toda la información del camino que se quiere

establecer. Esta solución da como resultado el par óptimo de caminos disjuntos extremo a extremo, pero tiene algunos inconvenientes. Dos de los más destacables son; que es necesario una ampliación estructural de la red para poder realizarlo, y que la compartición de información de routing entre dominios es una situación que los administradores suelen evitar. Es por esto que, aunque no sean óptimas, se barajan también otras posibles soluciones.

En este mismo trabajo [10] Ricciato et al. nos presenta un interesante análisis de algunas de estas propuestas con esquema distribuido para el cálculo de caminos disjuntos, además de un análisis de su efectividad en redes reales.

2.4.2. Métodos PPRO y ARO

Estos dos metodos distribuidos de cálculo y señalización de caminos disjuntos interdominio fueron las primeras propuestas serias de solución distribuida al problema del cálculo de caminos disjuntos interdominio.

El método PPRO utiliza dos fases para establecer los caminos, una por camino [12].

En la primera fase se calcula el camino principal y se señaliza. Para ello cada nodo de entrada a un dominio calcula el camino dentro de su dominio y lo señaliza hasta el nodo de salida. En este punto, el nodo de entrada de el dominio siguiente realiza la misma operación concatenando el camino que ha calculado con el camino que se ha señalizado hasta él. Esta operación se repite hasta el nodo final formando un camino principal extremo a extremo. El nodo final envía en el mensaje de respuesta RESV de RSVP un objeto RRO (*Record Route Object*).

La información contenida en este objeto será utilizada por el nodo inicial en la segunda fase. El contenido íntegro del objeto RRO se incluye en el contenido del nuevo objeto PPRO que se adjunta al mensaje de PATH de esta segunda fase, en la que se calcula y señaliza el camino de respaldo. El procedimiento para el calculo y señalización de este camino de respaldo es exactamente el mismo que en el caso del camino principal, excepto que en los calculos de los caminos internos a un determinado dominio se deben excluir todos aquellos nodos y SRLGs (*Shared Risk Link Group*) contenidos en el PPRO.

Como se explica en [10][74], e introduciremos en la sección 2.4.4, el problema es que es posible que la elección de ambos caminos de manera no simultánea lleve a no encontrar dos caminos disjuntos, aunque estos existan (topología trampa).

El método ARO (*Associated Route Object*) en cambio, propone otra forma de buscar estos caminos distribuidos disminuyendo la probabilidad de caer en una topología trampa [75]. En este método, los caminos disjuntos dentro de un dominio se calculan simultáneamente utilizando un algoritmo de búsqueda de caminos disjuntos

más cortos basado en la información proporcionada por el protocolo IGP (*Interior Gateway Protocol*) que se esté utilizando [9].

Con esta información se garantiza que el camino de respaldo disjunto existe dentro del dominio. Se señaliza el camino principal y se añade al objeto RSVP ARO el camino de respaldo calculado. Esta operación se realiza a la entrada de cada dominio que es necesario atravesar, hasta que se llega al nodo final. Por último, la información contenida en el objeto ARO es utilizada para señalizar el camino de respaldo.

Sin embargo, este esquema también adolece de algunos problemas, y es que, al igual que el método PPRO, puede caer en una topología trampa, haciendo necesario buscar caminos alternativos desde un dominio anterior, en el que ya se señalizó el camino principal. Hay que volver hacia atrás porque si el algoritmo de búsqueda de caminos disjuntos dentro de un dominio resuelve que estos no existen puede que sí que existan desde otros routers de entrada a la red. Esto obligaría a modificar los caminos del dominio anterior para que entren en el dominio actual por routers diferentes, desde los que puede que sí que se encuentren el par de caminos disjuntos. A este procedimiento se le denomina *crankback* [76], proceso que explicaremos más adelante y que, típicamente, está presente en las redes de conmutación de circuitos que requieren cierta reserva de recursos.

Terminaremos diciendo que la definición de el objeto PPRO forma parte de el estudio sobre señalización extremo a extremo que el IETF lleva a cabo en [12]. En este trabajo se definen elementos y procedimientos RSVP necesarios para la señalización de caminos GMPLS extremo a extremo en los distintos tipos de protección, 1+1, 1:1, 1:*n*, reencaminamiento, etc. La propuesta no es específica de caminos interdominio, se refiere a métodos de recuperación extremo a extremo con los caminos dados. Define cómo señalizar, cómo indicar qué tipo de protección se utilizará entre el camino principal y el camino de respaldo, etc. No indica nada de métodos o procedimientos para obtener dichos caminos disjuntos.

El estudio sobre el procedimiento de ARO está abandonado. El esfuerzo actual del IETF en la línea sobre cómo calcular dos caminos disjuntos que atraviesan diversos dominios se centra en la propuesta de mecanismos generales que permitan, basados en estos, ofrecer multitud de soluciones. Este cambio de filosofía es muy interesante porque permite que con la definición de pocos procesos y objetos se puedan proponer soluciones acordes a distintos escenarios particulares sin necesidad de modificar a cada nueva propuesta los protocolos involucrados (nuevos objetos RSVP, nuevos objetos IGP, etc).

Este cambio de visión es especialmente interesante en el ámbito interdominio, donde los escenarios pueden ser muy diversos. Por ejemplo, dependiendo de la cantidad de información que los dominios involucrados quieran o puedan compartir entre sí el procedimiento podrá ser más o menos sencillo y óptimo. No es lo mismo que los dominios se correspondan con áreas IGP, todas gestionadas por el mismo opera-

dor y donde se podrá compartir mucha información topológica, a que los dominios se correspondan con Sistemas Autónomos, con mayor o menor grado de confianza, pero entre los que no se revelan información topológica. Estos dos casos no deben compartir la misma solución de búsqueda de caminos disjuntos. Si fuese la misma solución ésta debería ser tan general que si fuese relativamente adecuada para ambos casos sería muy compleja y si no fuese lo suficientemente compleja no sería la más adecuada a cada caso.

En el siguiente apartado explicaremos algunos de estos objetos y procedimientos más generales que se están proponiendo y cómo estos sirven para definir soluciones similares a PPRO o ARO en el interdominio.

2.4.3. Objetos XRO y EXRS

La especificación de estos dos objetos y sus posibles usos se están definiendo actualmente en el IETF [77]. Primero los explicaremos y posteriormente daremos alguna pincelada sobre sus posibles usos.

El objeto RSVP XRO (*eXclude Route Object*) permite especificar una lista abstracta de nodos que no deben ser incluidos en un camino. Con el termino "*nodo abstracto*" se incluye a diferentes tipos de nodos, todos ellos válidos, tales como, direcciones IP, redes IP, SRLGs o, incluso, Sistemas Autónomos enteros.

Un nodo que recibe un mensaje de PATH con un objeto XRO no puede expandir un camino a otro que contenga nodos incluidos en dicho objeto XRO, o en el caso de no ser necesaria una expansión y sólo deba decidir el siguiente salto este no puede ser un nodo incluido en el objeto XRO[4].

El objeto RSVP EXRS (*Explicit eXclusion Route Subobject*) es un subobjeto del objeto ERO (*Explicit Route Object*). En este objeto se almacena una lista de nodos que deben ser evitados entre dos nodos, el anterior y posterior a él de la lista de nodos de la ruta explícita.

Se debe señalar que ambos objetos tienen una función específica, pero pueden utilizarse para diversos propósitos. Por ejemplo, estos objetos se pueden utilizar tanto para procedimientos intradominio como interdominio, puesto que es su definición no especifica un uso concreto.

Sirva como ejemplo que es posible utilizar el objeto XRO en lugar del objeto PPRO, antes descrito, con el mismo procedimiento de cálculo obteniendo la misma solución, puesto que, al fin y al cabo, el objeto PPRO lo único que contiene son los

[4]Hay excepciones en las que la elección puede incluir un nodo del objeto XRO. Cuando un nodo aparece en la lista de nodos a excluir con el bit L a 1, el nodo debería ser evitado, `SHOULD` en terminología IETF, pero no es obligatorio evitarlo, y por tanto, podría aparecer. En cambio, en el caso de ser 0 el bit L, el nodo tiene que ser evitado obligatoriamente, `MUST` en terminología IETF.

nodos del camino principal que deben ser evitados en la construcción del camino de respaldo, esto es, una lista de nodos que deben ser evitados.

En el siguiente apartado vamos a introducir las topologías trampas poniendo como ejemplo cómo estas pueden afectar estas al procedimiento de PPRO.

2.4.4. Topologías trampa

Con topología trampa nos referimos a aquellas en las que existiendo un par de caminos disjuntos entre dos pares de nodos el método de cómputo y señalización no es capaz de encontrarlos o necesita de varios intentos (por medio de procedimientos de *crankback*) para encontrarlos. En general un método cae en una trampa topológica por calcular de manera independiente el camino principal (i.e. PPRO) o por señalizar el camino antes de calcularlo completamente (i.e. ARO).

Una cierta topología puede ser una trampa para un esquema de búsqueda y no serla para otro, de hecho hay esquemas que intentan evitar estas trampas. Aquí, a modo de ejemplo, veremos el caso de una topología trampa para el esquema PPRO, mostrada en la figura 2.6 [10].

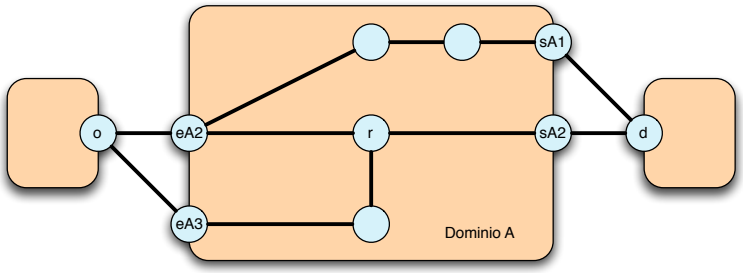


Figura 2.6: Ejemplo de topología trampa en el esquema PPRO

Si recordamos la metodología de PPRO, primero se calcula y establece el camino principal y una vez establecido éste se calcula, teniendo en cuenta los nodos y enlaces por los que pasa el camino principal, y establece el camino de respaldo. Fijémonos ahora en la figura, cuando se calcula el camino principal desde o a d el resultado de camino más corto es $eA2 - r - sA2$. Establecido este camino como camino principal no es posible calcular un camino disjunto para el camino de respaldo. Esto ocurre aún existiendo un par de caminos disjuntos entre o y d.

Una solución general al problema de las topologías trampa es utilizar procedimientos de *crankback*, los cuales permiten explorar más posibilidades volviendo atrás en los cálculos y probando con otros caminos. En nuestro ejemplo se volvería a calcular un camino principal que no fuese el calculado previamente. Si aún así diese

error se podría seguir intentando hasta un cierto número de veces o hasta explorar todas las posibilidades, lo que podría consumir demasiado tiempo y recursos.

2.4.5. Procedimiento de *crankback*

Puesto que el proceso de *crankback* no es la solución a un problema concreto de un determinado esquema es más correcto definir un procedimiento de *crankback* general que pueda ser utilizado por todas aquellas propuestas que precisen de él.

En [78] Farrel et al. explican el proceso de *crankback*, extendiendo RSVP-TE para que soporte esta nueva funcionalidad. El proceso de crankback genérico consiste en reencaminar un LSP desde un punto en el que ha habido un error a la hora de establecerlo. Este error puede deberse, por ejemplo, a que no hay recursos en el momento del establecimiento (si había durante el cálculo) o a que hay una porción del LSP que hace imposible encontrar un camino disjunto a los enlaces y nodos seleccionados para el camino de respaldo y hay que elegir una alternativa para poder encontrar un camino de respaldo válido.

2.5. Limitaciones de BGP en la búsqueda de caminos disjuntos interdominio extremo a extremo

Hasta ahora hemos considerado la búsqueda de caminos disjuntos utilizando la misma secuencia de Sistemas Autónomos desde un origen hasta un destino. Esto requiere el cálculo de caminos disjuntos dentro de cada AS. Una alternativa a este procedimiento sería que cada uno de los caminos llegase al destino por un AS_PATH distinto. De modo que siendo distinta la secuencia de AS atravesados los caminos son intrisecamente disjuntos. El problema radica en que utilizando BGP como protocolo de routing inter-AS esto no es posible.

BGP en lugar de anunciar en la red un destino y el coste (distancia) de alcanzarlo anuncia una red y una secuencia de ASes (AS_PATH) para alcanzar dicha red destino. Por eso se dice que BGP es un procolo de routing de tipo *path-vector*.

Como BGP no muestra información de routing interno, utilizar BGP no garantiza que se utilice el camino más corto a un destino. En general, BGP minimiza el numero de ASes que es necesario atravesar, independientemente del número de saltos interno, retardo extremo a extremo o ancho de banda de las líneas.

Las políticas de uso de BGP pueden ser utilizadas para influir en la selección de las rutas, pero en general, de todas las rutas para un determinado destino que llegan a un router de borde de un dominio de los routers de borde de los dominios vecinos

BGP, este debe escoger la de menor número de ASes y desechar el resto. Esto significa que los routers de borde de un dominio sólo conocen una forma de alcanzar un determinado destino. Este hecho impide poder utilizar BGP para realizar ingeniería de tráfico interdominio. La información disponible a nivel BGP es muy escasa y limitada. En particular, no es posible calcular caminos alternativos o caminos con ciertas restricciones. La única ingeniería que se puede hacer es de tipo local [79] (e.g. el *Multi Exit Discriminator* permite seleccionar el router de entrada para un destino y el atributo *Local_preference* permite seleccionar el router de salida).

Por eso, si se quiere utilizar una estrategia de búsqueda de caminos disjuntos que atraviesen ASes disjuntos sería necesaria una modificación del protocolo BGP [80][81]. En [82] se puede ver un ejemplo de cómo se podría aprovechar esta utilizando PCE.

Capítulo 3

Planteamiento del problema y objetivos

En este capítulo se presenta el problema que estudia esta Tesis Doctoral, mostrando los escenarios donde se va a trabajar así como los inconvenientes de las soluciones propuestas hasta ahora. Para finalizar se plantearán los objetivos que se pretenden cubrir en este trabajo.

3.1. Problema a resolver

La protección de recursos de red utilizando MPLS ha sido muy tratada en la literatura e implantada en las redes reales, pero siempre referidas a recursos dentro de una misma red, esto es a recursos que dependen del mismo administrador.

El problema, al que se pretende dar solución, surge cuando lo que se quiere proteger son recursos del interdominio, es decir, recursos utilizados para conectar entre sí dominios de gestión administrativa distintos. Las soluciones propuestas deberán ser simples, viables y escalables.

Hay dos motivos principales por los que no es trivial proteger estos recursos. Primero, no siempre es factible computar un camino alternativo por otros dominios que evite el dominio en el que han fallado los recursos. A nivel de Sistemas Autónomos, esto se debe a que el protocolo de encaminamiento interdominio utilizado, BGP, es un protocolo que informa de caminos entre ASes, no del grafo de dominios, por lo que no se dispone de un camino alternativo entre estos ASes. Segundo, por estar involucradas redes de gestión administrativa independiente ocurre que desde una red no se tiene conocimiento del grafo de la red del resto de dominios involucrados. Es por este motivo que un dominio no es capaz de computar un camino alternativo a otro camino de forma que ambos transcurran por el mismo dominio. Esta limita-

ción lleva a utilizar mecanismos de computación distribuida. Aún así hay que tener cuidado con la información compartida, así como con el tiempo de convergencia de cómputo de estas técnicas distribuidas y la dificultad/complejidad de implantarlo en una red.

Generalmente, cuando en la literatura se habla de protección de recursos en una red esta suele venir acompañada de un análisis de la capacidad extra que hay que reservar en la red para el sistema de protección. Fundamentalmente se analiza cuánto ancho de banda extra es necesario en la red para protegerla del fallo de uno de los enlaces o nodos de la misma. Estos esquemas suelen compartir la capacidad reservada para protección en un enlace entre varios enlaces protegidos. Típicamente, en este análisis se compara la capacidad necesaria del esquema propuesto con otros esquemas existentes para proteger toda una red y concluir qué esquemas consumen más recursos y cuáles menos [52].Esto está referido a la protección de enlaces dentro de un dominio.

En contraste, esta Tesis Doctoral se centra específicamente en el cómputo y establecimiento de caminos disjuntos de ingeniería de tráfico en redes donde se ha reservado en todos los enlaces una capacidad suficiente para los LSPs de respaldo. En otras palabras, no se trata de hacer CbR (*Constraint-based Routing*) al uso, en el que típicamente se buscan caminos con una determinada capacidad residual, ya que cualquier mecanismo de este tipo es susceptible de ser empleado para que un operador obtenga información de la ocupación/capacidad de las rutas internas de un competidor. Por consiguiente, nos centramos en un tipo de CbR donde la restricción es exclusivamente la calidad de disjunto de un camino respecto a otro, y se pretende realizar en base sólo a información de encaminamiento existente o con modificaciones mínimas en los esquemas de encaminamiento IGP o EGP (*Exterior Gateway Protocol*).

En definitiva, la presente Tesis Doctoral se centra en evolucionar el actual sistema encaminamiento en redes IP *best effort*, como Internet, para proporcionar reencaminamiento rápido a agregados de tráfico concretos gracias a MPLS. Esto supone un avance sobre el sistema de encaminamiento actual, el cual reacciona en caso de fallos en la red proporcionando rutas alternativas, con tiempos de recuperación no acotados, sin asegurar que la nueva ruta mantendrá sus prestaciones.

En el siguiente apartado se muestran los escenarios sobre los que se resolverá en esta Tesis Doctoral la problemática planteada.

3.2. Escenarios de uso

En la figura 3.1 se muestra el escenario general sobre el que se desarrolla esta Tesis Doctoral. Como puede verse consiste en una serie de dominios con BGP y un

IGP interconectados entre sí. Siempre hay un dominio origen (Dominio 1 en la figura) y un dominio destino (Dominio 7 en la figura) que albergan sendos LSRs, LSR O y LSR D, entre los que se quiere establecer un LSP principal y uno de respaldo que lo proteja. La aplicación principal de esta funcionalidad será ofrecer "reencaminamiento rápido" para tráfico sensible a cortes de transmisión como puede ser el tráfico de voz entre pasarelas, enlaces sobre Internet, circuitos conmutados, o sistemas de tiempo real crítico.

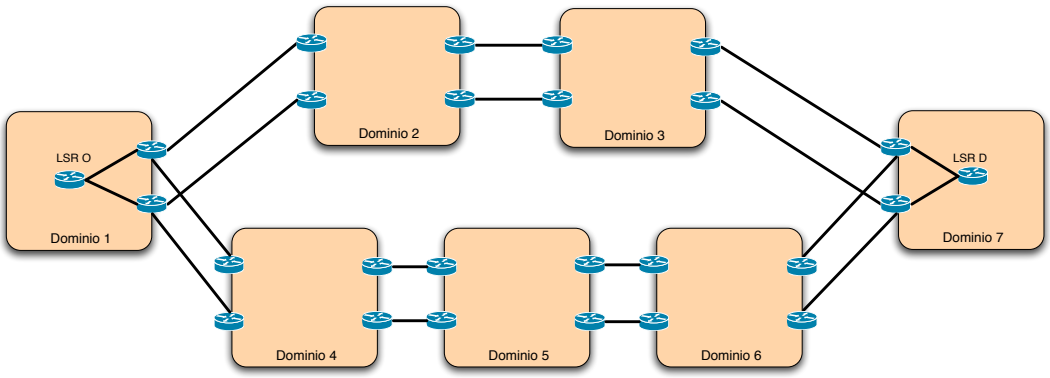


Figura 3.1: Escenario general con múltiples AS_PATHs.

El dominio de origen no tiene porqué conocer la topología completa de dominios, por ejemplo, si se utiliza BGP como EGP, sólo conocerá un camino de dominios (AS_PATH) entre origen y destino. A su vez, un router de un dominio no tiene porqué conocer la topología interior del dominio al que pertenece, por ejemplo, si se utilizan distintas áreas OSPF como IGP sólo conocerá la topología de su área, no de todo el dominio.

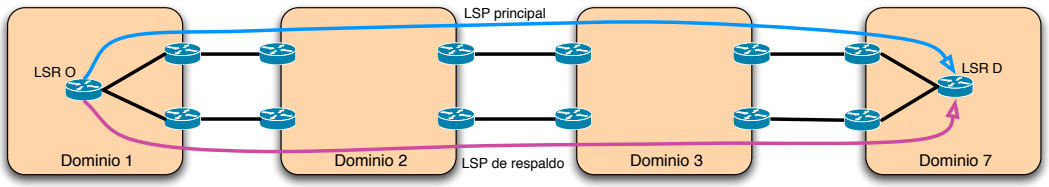


Figura 3.2: Escenario con un único AS_PATH.

Dividiremos el problema en dos casos; primero, cuando sólo se conoce un AS_PATH y por tanto hay que localizar dos caminos disjuntos entre origen y destino que compartan dominios (figura 3.2); y segundo, cuando se conocen dos AS_PATHs disjuntos entre origen y destino y hay que computar un camino por cada uno de los

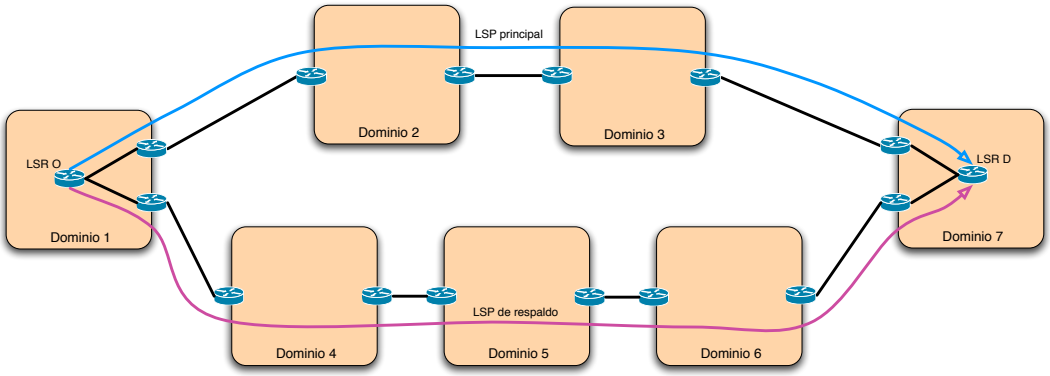


Figura 3.3: Escenario con dos AS_PATHs.

AS_PATHs (figura 3.3). Este último es un supuesto muy interesante no estudiado en la literatura.

A continuación se analizan los inconvenientes de las soluciones actuales para poder estudiar analizar qué virtudes se requieren de las soluciones buscadas y plasmarlas en los objetivos generales de esta Tesis Doctoral.

3.3. Inconvenientes de las soluciones actuales

En contenido de este apartado es una visión desde otro punto de vista de los inconvenientes, ya comentados en el apartado 2.4, de las soluciones dadas en la literatura al problema aquí planteado.

Fundamentalmente existen dos tipos de soluciones propuestas hasta ahora, las distribuidas o “por dominio”, como ARO y PPRO, y “basada en PCE”.

En [83] se analizan estos dos tipos de aproximación al computo de LSPs interdominio, “por dominio” y “basada en PCE”. En dicho artículo Dasgupta y cia. justifican la utilización de PCE debido a los problemas de la aproximación “por dominio”. En este trabajo se analizan los problemas de ambos esquemas en el cómputo de caminos que requieran reserva de recursos, no en la búsqueda de caminos disjuntos, pero en ambos casos el análisis es similar y por ello válido. En [10] también se realiza un análisis en este sentido, pero centrado en el cómputo de caminos disjuntos. Fundamentalmente, los problemas en los esquemas distribuidos vienen de la necesidad de utilizar técnicas de *crankback*, que puede provocar un aumento de procesamiento en los nodos intermedios, implicándolos en una búsqueda y una señalización que puede ser infructuosa, así como una alta variabilidad en el tiempo de cómputo de los caminos. Esta variabilidad en el tiempo de cómputo, asociada con el *crankback*, implica la necesidad de limitar el tiempo de *crankback*, con la

posibilidad de no disponer de una solución aún existiendo [83].

La utilización de técnicas de *crankback* se ha venido justificando como un mecanismo necesario para evitar las topologías trampa [10][78]. Cuando en una red hay recursos agotados que no pueden cursar tráfico el grafo de red a la hora de computar caminos se ve modificado, de modo que haya que evitar estos recursos. Este hecho hace que el grafo de red sobre el que hay que computar caminos no sea el grafo original de la red, y donde era difícil encontrar una topología trampa aparezca una fácilmente. Por este motivo, es importante ser capaz de evitar las topologías trampa, porque, si bien, es posible que no existan en el grafo completo de una red si puedan aparecer cuando hay que tener en cuenta requerimientos de calidad de servicio (QoS) o cuando al grafo de red se le añade el concepto de SRLG (véase el apéndice A), ya que en ambos casos el grafo se convierte en un subgrafo de la red.

El principal problema de PCE es la necesidad de implantar y gestionar infraestructura adicional en cada uno de los dominios. Con lo que si se utiliza este tipo de solución, para realizar LSPs interdominio todos los dominios involucrados están obligados a realizar modificaciones importantes en su red y en la forma de planificarla. Además, como se ha citado, es improbable que los operadores estén dispuestos a federar elementos que pueden desvelar las capacidades de sus rutas, ya que pese a plantearse como objetivo de diseño la posibilidad de ocultar detalles internos de la red en las consultas, la sola respuesta afirmativa o negativa a una consulta permite inferir información. Otro problema secundario es que debido a que el elemento que realiza la computación no son los nodos de red en sí mismos, si no en un elemento externo (PCE), es posible que desde que se computó el LSP hasta que se señaliza algún recurso se haya ocupado y se requiera una recomputación del LSP, luego, en algún caso, puede llegar a ser necesario utilizar técnicas de *crankback* entre PCEs para recalcular el camino, cayendo, aunque en menor medida, en los problemas antes mencionados, pudiéndose convertir, además, en puntos únicos de fallo.

Otro problema importante a tener en cuenta y que en [83] se soslaya es que el retardo en computar y señalizar los LSPs utilizando PCE es mayor que en otras técnicas, como por ejemplo las “combinadas”, en las que el cómputo y la señalización se realizan en un sólo paso [74]. Se requiere de tres pasos, uno de computación, otro de establecimiento, y otro de resolución de “token”, para todo el proceso. En un primer paso, la petición de cómputo se propaga de PCE en PCE desde el PCE del AS origen al de destino para volver al nodo origen con el cálculo del camino realizado. Una vez que el origen tiene el camino calculado inicia la fase de señalización (segundo paso), pero en cada LSR de entrada a un origen se debe consultar al PCE local cuál es el camino a utilizar dentro del dominio (tercer paso). Este último paso es necesario si se utilizan identificadores opacos de caminos (o “tokens”) para ser compartidos entre dominios y así preservar la privacidad entre dominios.

En los esquemas distribuidos o “por dominio”, como ARO o PPRO, el cómputo

y señalización se pueden realizar en un único paso o a lo sumo en dos. La única ventaja real de PCE es la obtención de la solución óptima con el coste de una mayor complejidad, a costa de incluir infraestructura importante adicional en la red.

En base a estos inconvenientes y los escenarios de uso se han elaborados los objetivos generales de esta Tesis Doctoral.

3.4. Objetivos

El objetivo principal de esta Tesis Doctoral es *estudiar los retos técnicos de los sistemas de protección en escenarios de red IP-MPLS multidominio y proponer soluciones simples y viables en la práctica a los mismos*. Este objetivo principal se concreta en los siguientes objetivos particulares:

- Encontrar un algoritmo de búsqueda de caminos disjuntos interdominio que evite las topologías trampa, evitando costosas soluciones de tipo PCE.
- Procurar que el algoritmo a su vez no necesite un mecanismo de *crankback* para conseguir su proposito en el 100 % de los casos posibles.
- Definir un mecanismo de señalización completo para el establecimiento de los caminos disjuntos, principal y respaldo, que contemple el algoritmo propuesto. Para el caso interdominio y el caso interárea.
- Proponer una modificación de BGP que permita computar AS_PATHs disjuntos que permita computar los caminos principal y de respaldo disjuntos de un modo sencillo.
- Estudiar una alternativa PCE, proponiendo mecanismos distribuidos que sean más eficientes que PCE en el intercambio de información y tiempo de establecimiento, cumpliendo con los requisitos de aislamiento y privacidad de los dominios involucrados.
- Analizar las posibilidades reales de utilizar p-ciclos para realizar protección multidominio.

Parte II

Propuestas de mecanismos de soporte a la protección multidominio

Capítulo 4

Cómputo de caminos disjuntos entre pares de nodos en el interior de un dominio

En este capítulo se va a explicar y demostrar el funcionamiento de un algoritmo de búsqueda de caminos disjuntos entre dos pares de nodos. En nuestro caso se tratará de dos nodos de entrada a un dominio y dos nodos de salida del mismo dominio, pero podrían ser cuatro nodos cualesquiera. Tras una introducción al problema que se intenta resolver y una exposición de la notación que se utilizará posteriormente, se presenta el algoritmo y su funcionamiento. Tras unos ejemplos ilustrativos se demostrará formalmente que, cumplidas las condiciones de partida, siempre se obtiene una solución válida mediante el algoritmo propuesto. Analizando las propiedades del algoritmo se demostrará que éste obtiene una solución siempre que ésta exista, incluso cuando no se cumplan las condiciones de partida antes mencionadas. Seguidamente se realiza un breve análisis de la posible generalización del algoritmo para más de dos pares de nodos. En las conclusiones finales se destacarán las propiedades del algoritmo.

En el capítulo 5 se presentará un mecanismo de protección multidominio basado en la utilización de este algoritmo.

4.1. Introducción

Antes de abordar el cómputo de caminos disjuntos extremo a extremo que pasen por varios dominios en los siguientes capítulos, es preciso abordar el cálculo de caminos disjuntos dentro de un único dominio, desde un par de nodos de entrada a un par de nodos de salida. Para ello suponemos un dominio cualquiera, como por

ejemplo, el dominio K de la figura 4.1. El objetivo es obtener un par de caminos disjuntos que unan los nodos de entrada, e_1^k y e_2^k, con los nodos de salida, s_1^k y s_2^k, de manera distribuida, es decir, no se supone la existencia de un PCE que compute el par de caminos óptimo, y teniendo en cuenta que cada router puede tener una visión distinta de la red. Cada nodo de entrada puede pertenecer a un área IGP distinta[1].

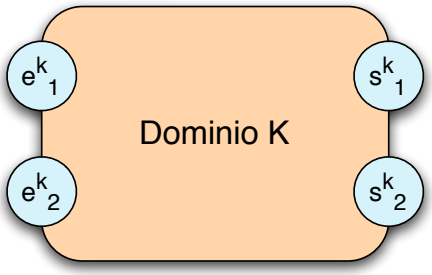


Figura 4.1: Dominio con dos nodos de entrada y dos nodos de salida

Dados estos dos pares de nodos pueden darse dos conjuntos de soluciones; primera, dos caminos disjuntos, uno entre e_1 y s_1 ($P^s_{\{e_1,s_1\}}$) y otro entre e_2 y s_2 ($P^s_{\{e_2,s_2\}}$); y segunda, uno entre e_1 y s_2 ($P^s_{\{e_1,s_2\}}$) y otro entre e_2 y s_1 ($P^s_{\{e_2,s_1\}}$). En la figura 4.2 se muestra una representación gráfica de estos caminos, caminos que, en realidad, consisten en una secuencia de nodos y enlaces desde uno de los nodos de entrada hasta uno de los nodos de salida[2].

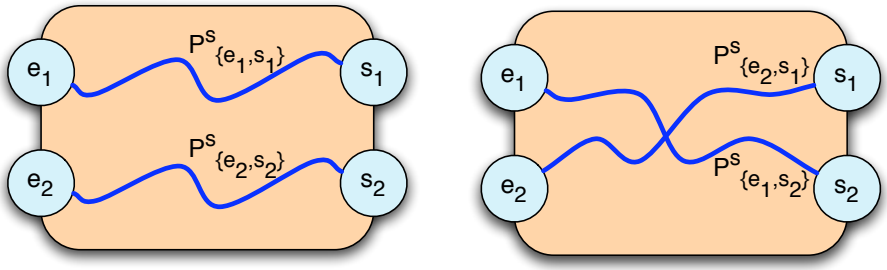


Figura 4.2: Soluciones posibles

El algoritmo presentado a continuación es capaz de determinar una de las dos soluciones posibles presentadas anteriormente a partir de la información proporcio-

[1] A partir de este momento, cuando sólo haya un dominio y no haya lugar a la ambigüedad se utilizará e_i (s_i) en lugar de e_i^k (s_i^k).

[2] A partir de este momento, por simplicidad de notación, se utilizará $P_{1,2}$ como alternativa de $P_{\{e_1,s_2\}}$ en los contextos donde no haya ambigüedad.

nada, de manera independiente, por los nodos de entrada al dominio. Esta información, por parte de cada uno de los nodos de entrada, no es más que un par de caminos disjuntos hasta cada uno de los nodos de salida (véase la figura 4.3). Por tanto, $P_{1,1}$ y $P_{1,2}$ son dos caminos disjuntos desde e_1 hasta s_1 y s_2 respectivamente. De igual modo, $P_{2,1}$ y $P_{2,2}$ son dos caminos disjuntos desde e_2 hasta s_1 y s_2 respectivamente.

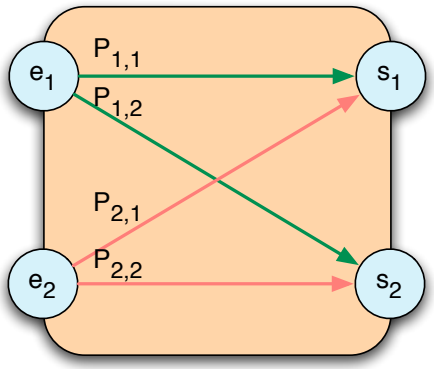


Figura 4.3: Dos pares de caminos disjuntos, un par desde cada nodo de entrada a los nodos de salida

A partir de estos cuatro caminos, facilitados de manera independiente por los nodos de entrada, y por medio del algoritmo que se va a presentar es posible computar dos caminos disjuntos entre los nodos de entrada y salida del dominio (figura 4.2).

Antes de explicar el algoritmo se van a detallar cuáles son los requisitos previos del mismo para poder ser aplicado.

4.2. Requisitos previos

Ya se ha explicado el objetivo del algoritmo; a partir de la información que de manera independiente proporciona cada nodo de entrada de un dominio a la entidad que ejecuta el algoritmo éste computa dos caminos disjuntos entre los nodos de entrada y salida de dicho dominio. La información que proporciona cada uno de los nodos de entrada es un par de caminos disjuntos desde ese nodo hasta los nodos de salida (i.e. $P_{1,1}$ y $P_{1,2}$ para el nodo e_1 en el ejemplo de la figura 4.3).

Para que los nodos de entrada puedan calcular estos caminos deben disponer del grafo de red de todo el dominio. Si en el dominio existen herramientas de ingeniería de tráfico, y así debe ser para poder computar y crear caminos de respaldo, se utilizará un protocolo de encaminamiento basado en estado de enlaces, como OSPF o IS-IS. Estos protocolos proveen a todos los nodos de la red la información topológica

necesaria.

Como ya se aclaró en el capítulo de introducción, *dominio* aquí se refiere a un conjunto de elementos de red que forman parte del mismo contexto administrativo, como, por ejemplo, un Sistema Autónomo o un área IGP.

Disponiendo del grafo del dominio cada nodo de entrada, éste sólo necesita ejecutar un algoritmo de búsqueda de caminos disjuntos, como el propuesto por Suurballe [9], para obtener los caminos disjuntos a los nodos de salida.

Dados estos cuatro caminos el algoritmo es capaz de computar dos caminos disjuntos entre los dos pares de nodos, entrada y salida. Por tanto, siempre que sea posible computar $P_{1,1}$, $P_{1,2}$, $P_{2,1}$ y $P_{2,2}$ será posible encontrar ese par de caminos disjuntos entre los nodos de entrada y los de salida. La pregunta que cabe hacerse ahora es cuándo es posible encontrar aquellos.

En el apéndice A hay desarrollado un pequeño estudio sobre redes k-conectadas. En el caso particular de las redes 2-conectadas podemos definirlas de la siguiente manera.

Definición 4.1. *Una red es 2-conectada si el grafo de la red permanece conectado si se eliminan menos de 2 nodos del grafo.*

Una consecuencia directa de esta definición es que:

Definición 4.2. *Una red 2-conectada es una red en la que entre todo par de nodos de la red existen al menos dos caminos disjuntos que los unen.*

Proposición 4.1. *Si se añade un nuevo nodo conectado a dos nodos de una red 2-conectada la nueva red sigue siendo 2-conectada.*

Demostración. La red de la figura 4.4 se muestra una red 2-conectada a la que se le ha añadido un nodo nuevo, *n*, conectado a dos nodos, 1 y 2, de la red 2-conectada.

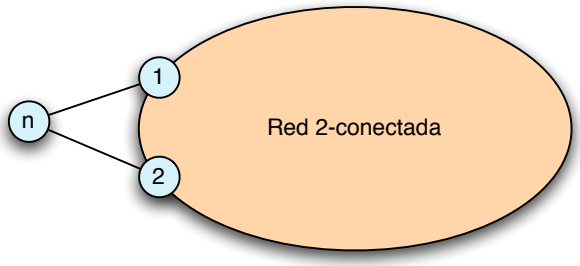


Figura 4.4: Añadiendo un nuevo nodo a una red 2-conectada.

Como se ha dicho en la definición 4.1 una red 2-conectada lo es si al eliminar un nodo la red sigue conectada. Se probarán todas las posibilidades de los nodos implicados:

1. Si se elimina el nodo añadido n sigue quedando la red original, que es 2-conectada y, por tanto, la red sigue conectada.
2. Si se elimina el nodo 1 la red original sigue conectada puesto que era 2-conectada y al eliminar un nodo no se desconecta y el nodo nuevo n sigue conectado a la red por medio de 2. Por tanto, la red, después de eliminar el nodo 1, sigue conectada
3. Si se elimina el nodo 2 estamos en el mismo caso anterior.

Eliminando cualquiera de los tres nodos involucrados en el "crecimiento de la red" esta sigue manteniéndose conectada, por tanto, la red resultante es 2-conectada.

□

Lema 4.1. *En una red 2-conectada siempre es posible encontrar* $P_{\{e_1,s_1\}}$ *y* $P_{\{e_1,s_2\}}$ *disjuntos entre sí, y* $P_{\{e_2,s_1\}}$ *y* $P_{\{e_2,s_2\}}$ *también disjuntos entre sí.*

Demostración. Para la demostración basta con demostrar que desde un nodo cualquiera, e.g. e_1, es posible encontrar dos caminos disjuntos a otros dos nodos cualesquiera, e.g. s_1 y s_2, si la red es 2-conectada.

En la figura 4.5 se puede ver un esquema de la situación planteada. Se ha añadido además el nodo n de modo que el conjunto sea 2-conectado (véase la proposición 4.1).

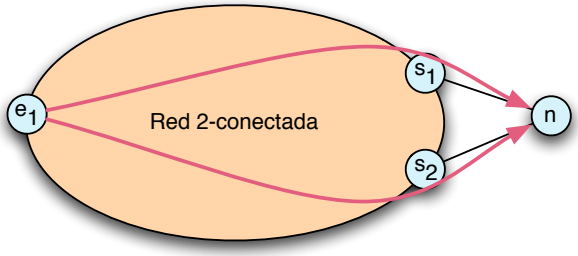


Figura 4.5: Caminos disjuntos en una red 2-conectada.

Según la definición 4.2 entre todo par de nodos de una red 2-conectada es posible establecer dos caminos disjuntos. En particular, es posible hacerlo entre e_1 y n. Como s_1 y s_2 son los dos únicos nodos que conectan a n con el resto de la red cada uno

de estos caminos disjuntos deberá pasar por uno de los nodos, y sólo uno de ellos. Tenemos entonces dos caminos disjuntos que comienzan en e_1 y uno de ellos pasa por s_1 y el otro por s_2.

□

Por tanto una red 2-conectada es condición suficiente (no es necesaria) para poder utilizar el algoritmo, puesto que es una red sobre la que se puede calcular los caminos de partida.

Parece razonable pensar que una red sobre la que se quiere realizar protección de enlaces y nodos, tanto a nivel intradominio como interdominio, sea suficientemente redundante como para poder proveer caminos alternativos entre todos los nodos, es decir, que al menos sea 2-conectada.

A continuación, se explica la notación utilizada, tanto en el algoritmo y la demostración que se detallan a continuación, como en la generalización del algoritmo a N nodos que se realiza posteriormente.

4.3. Notación utilizada

- N_e es el número de nodos de entrada del dominio.
- N_s es el número de nodos de salida del dominio.
- $P_{i,j} =< i, n_2, n_3, \ldots, n_{m-1}, j >$ es el camino entre el nodo de entrada i y el nodo de salida j.
- $P_{i,j} \perp P_{x,y}$ indica que los caminos $P_{i,j}$ y $P_{x,y}$ no tienen ningún elemento en común, con la posible excepción del primer y/o último elemento, es decir, puede ocurrir que $i = x$ y/o $j = y$. Comiencen o finalicen en el mismo nodo estos caminos no se cortan en ningún otro punto, es lo que se denominará caminos disjuntos.
- $P_{i,j} \not\perp P_{x,y}$ indica que los caminos $P_{i,j}$ y $P_{x,y}$ tienen algún elemento intermedio en común, es decir, no son disjuntos.
- $P^s_{i,j}$ es un camino solución que une el nodo de entrada i con el nodo de salida j.
    - Como máximo un camino solución puede salir de cada nodo de entrada y como máximo un camino solución puede llegar a cada nodo de salida.
    - Además, todos los $P^s_{i,j}$ son disjuntos entre sí (por eso son solución).

- Se denota un conjunto genérico de soluciones como $P^s_{\left\{\begin{smallmatrix}1\\ \cdot\\ \cdot\\ N_e\end{smallmatrix}\right\},\left\{\begin{smallmatrix}1\\ \cdot\\ \cdot\\ N_s\end{smallmatrix}\right\}}$.

  Por ejemplo, en el caso de dos nodos de entrada ($N_e = 2$) y dos nodos de salida ($N_s = 2$), como en la figura 4.3, hay dos conjuntos de soluciones posibles, $P^s_{1,1}$ y $P^s_{2,2}$ ó $P^s_{1,2}$ y $P^s_{2,1}$ (ver figura 4.2). $P^s_{\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}}$ se refiere a cualquiera de ellas.

- Los n caminos solución también se pueden denotar como $\langle P_1, \ldots, P_n \rangle^s$ en el caso de que el punto de origen y destino sea indiferente. Así por tanto en el caso de que sean dos los caminos solución, estos vendrán indicados por $\langle P_1, P_2 \rangle^s$.

- Si los caminos $P_{i,j}$ y $P_{x,y}$ no son disjuntos se denota el primero de sus cruces como $\overset{P_{i,j}}{\underset{P_{x,y}}{\oplus}}$. Se entiende por primer cruce como el primer nodo o enlace[3] común a ambos caminos comenzando por i.

- Dados $P_{i,j}$ y $P_{x,y}$ no disjuntos entre sí, $\left[P_{i,j} \overset{P_{i,j}}{\underset{P_{x,y}}{\oplus}} P_{x,y}\right]$, o de modo simplificado, $P_{i,j \oplus x,y}$, denota el camino construido por la concatenación del segmento constituido por $P_{i,j}$ desde su inicio hasta su cruce con $P_{x,y}$ y el segmento dado por $P_{x,y}$ desde su cruce con $P_{i,j}$ hasta el final. Es un camino, por tanto, que comienza en i y termina en y, $P_{i,y}$.

4.4. Algoritmo propuesto. Algoritmo de las Parejas Disjuntas.

En este apartado se presentará un algoritmo para el cálculo de caminos disjuntos entre dos nodos de entrada, $N_e = 2$, y dos nodos de salida, $N_s = 2$, dentro de un dominio. Obtiene $P^s_{\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}}$ a partir de los $P_{i,j}$ proporcionados por los nodos de entrada. Este algoritmo se denomina APD (*Algoritmo de las Parejas Disjuntas*).

En los próximos capítulos de esta Tesis Doctoral se verán algunas de las posibles aplicaciones de este algoritmo, pero el escenario que lo motivó se describe a continuación. En una situación multidominio, existe un dominio X que necesita que un camino principal y uno de respaldo pasen por un dominio K, estos deben ser disjuntos dentro de este dominio. El dominio X tiene constancia de la existencia de e_{k1} y

[3]El primer elemento en común de dos caminos puede ser un enlace, aunque no provengan del mismo nodo o terminen en el mismo nodo, ya que pueden ir por la misma zanja física y compartir el mismo SRLG.

e_{k2}, pero no del grafo interno del dominio K, grafo que además K no va a facilitar a X. Si no hay presencia de PCEs en los diferentes dominios, el cómputo debe ser distribuido, pero ya se ha visto que en los cómputos distribuidos existentes es fácil caer en trampas topológicas (sección 2.4.4).

Si e_1^k y e_2^k calculan de manera independiente dos caminos disjuntos, uno a cada uno de los nodos de salida, s_1^k y s_2^k, cualquier nodo que disponga de estos cuatro caminos utilizando este algoritmo es capaz de obtener siempre dos caminos disjuntos entre los nodos de entrada y salida del dominio K.

La información de entrada al algoritmo son $P_{e_1^k,s_1^k}$, $P_{e_1^k,s_2^k}$, $P_{e_2^k,s_1^k}$ y $P_{e_2^k,s_2^k}$, o, simplemente, $P_{1,1}$, $P_{1,2}$, $P_{2,1}$ y $P_{2,2}$ tal que $P_{1,1} \perp P_{1,2}$ y $P_{2,1} \perp P_{2,2}$. Con esta información, por medio del algoritmo, obtendremos $P^s_{\{1\}\{1\}}_{\{2\}\{2\}}$.

El algoritmo APD consiste en dos pasos:

1. De entre las parejas de caminos

$$\langle P_{1,1},\ P_{2,2}\rangle \tag{4.1}$$

$$\langle P_{1,2},\ P_{2,1}\rangle \tag{4.2}$$

   y escoger aquellas que sean disjuntas.

   a) Si ninguna es disjunta ir al paso 2.

   b) Elegir la de menor coste[4]. El algoritmo termina.

2. Si existe el correspondiente cruce, computar las siguiente parejas de caminos

$$\langle P_{1,1},\ [P_{2,2 \oplus 1,2}]\rangle \tag{4.3}$$

$$\langle P_{1,2},\ [P_{2,2 \oplus 1,1}]\rangle \tag{4.4}$$

$$\langle [P_{1,1 \oplus 2,2}],\ P_{2,1}\rangle \tag{4.5}$$

$$\langle [P_{1,1 \oplus 2,1}],\ P_{2,2}\rangle \tag{4.6}$$

   y escoger aquellas que sean disjuntas.

   a) Elegir la de menor coste. El algoritmo termina.

[4]El coste puede estar definido de diversas maneras, i.e. el conjunto de los dos caminos sea el de menor número de saltos o aquel en el que el camino principal sea el de menor número de saltos

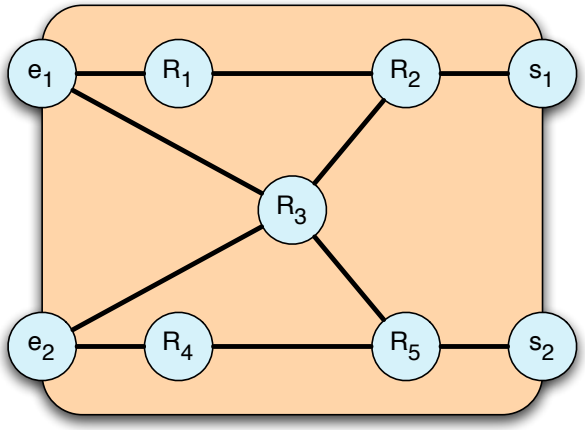


Figura 4.6: Ejemplo 1 de utilización del algoritmo APD. Paso 1.

Teorema 4.1. *Si se disponen de un par de caminos disjuntos, $P_{1,1}$ y $P_{1,2}$, desde un nodo, e_1, hasta otros dos nodos, s_1 y s_2, y de otro par de caminos disjuntos, $P_{2,1}$ y $P_{2,2}$, desde otro nodo, e_2, hasta los mismos dos nodos anteriores, s_1 y s_2, entonces siempre es posible obtener un par de caminos desde e_1 y e_2 hasta s_1 y s_2 disjuntos entre sí.*

La demostración de este teorema y del correcto funcionamiento del algoritmo APD se realiza en la sección 4.4.2. Previamente se muestran algunos ejemplos de funcionamiento del algoritmo.

4.4.1. Ejemplos

En este apartado se muestran algunos ejemplos de funcionamiento del algoritmo APD.

Ejemplo 1 En la figura 4.6 se puede ver una red simple utilizada para mostrar un sencillo ejemplo de uso del algoritmo APD.

Del grafo de la red se puede ver que los caminos disjuntos computados por e_1 y e_2 serían
$P_{1,1} = e_1 - R_1 - R_2 - s_1$
$P_{1,2} = e_1 - R_3 - R_5 - s_2$

$P_{2,1} = e_2 - R_3 - R_2 - s_1$
$P_{2,2} = e_2 - R_4 - R_5 - s_2$

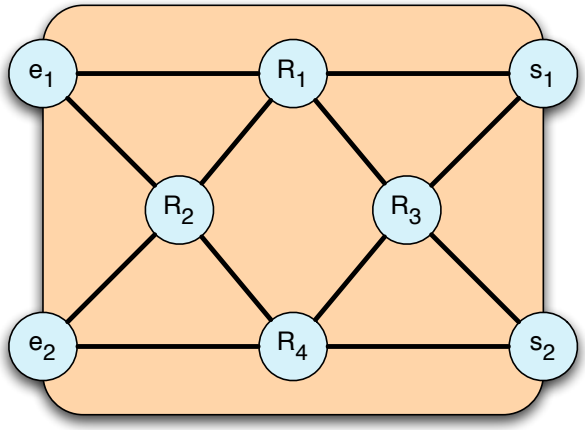


Figura 4.7: Ejemplo 2 de utilización del algoritmo. Pasos 1 y 2.

de tal modo que $P_{1,1}$ y $P_{1,2}$ sean disjuntos ($P_{1,1} \perp P_{1,2}$), y $P_{2,1}$ y $P_{2,2}$ también disjuntos ($P_{2,1} \perp P_{2,2}$).

Ejecutando el algoritmo y empezando por el paso 1, tenemos las parejas:

Pareja 1 $(P_{1,1}, P_{2,2}) = (e_1 - R_1 - R_2 - s_1,\ e_2 - R_4 - R_5 - s_2)$

Pareja 2 $(P_{1,2}, P_{2,1}) = (e_1 - R_3 - R_5 - s_2,\ e_2 - R_3 - R_2 - s_1)$

La pareja 2 no es disjunta, con lo que ésta se descarta. De modo que sólo queda la pareja 1, por lo que la solución del algoritmo es directamente la pareja 1.

Solución:

- $P^s_{1,1} = P_{1,1} = e_1 - R_1 - R_2 - s_1$
- $P^s_{2,2} = P_{2,2} = e_2 - R_4 - R_5 - s_2$

Ejemplo 2 Con ayuda de la red de la figura 4.7 se presenta ahora un ejemplo más de utilización del algoritmo APD que requiere ir al paso 2.

En este caso los caminos disjuntos calculados por e_1 y e_2 son
$P_{1,1} = e_1 - R_1 - R_3 - s_1$
$P_{1,2} = e_1 - R_2 - R_4 - s_2$

$P_{2,1} = e_2 - R_2 - R_1 - s_1$
$P_{2,2} = e_2 - R_4 - R_3 - s_2$

Aunque estos no sean los caminos disjuntos óptimos dado el grafo de red de la figura, la condición de disjuntez, unido a condiciones de QoS o políticas locales en e_1 y e_2 puede llevar a que el cómputo resulte en los caminos propuestos en el ejemplo.

Ejecutando el algoritmo y empezando por el paso 1 se tienen las parejas:

Pareja 1 $(P_{1,1}, P_{2,2}) = (e_1 - R_1 - R_3 - s_1,\ e_2 - R_4 - R_3 - s_2)$

Pareja 2 $(P_{1,2}, P_{2,1}) = (e_1 - R_2 - R_4 - s_2,\ e_2 - R_2 - R_1 - s_1)$

Ninguna de las parejas es disjunta, luego ninguna sirve como solución. Se debe ejecutar el paso 2 del algoritmo, lo que hace calcular las parejas:

Pareja 3 $(P_{1,1}, P_{2,2 \oplus 1,2}) = (e_1 - R_1 - R_3 - s_1,\ e_2 - \boldsymbol{R_4} - s_2)$

Pareja 4 $(P_{1,2}, P_{2,2 \oplus 1,1}) = (e_1 - R_2 - R_4 - s_2,\ e_2 - R_4 - \boldsymbol{R_3} - s_1)$

Pareja 5 $(P_{1,1 \oplus 2,2}, P_{2,1}) = (e_1 - R_1 - \boldsymbol{R_3} - s_2,\ e_2 - R_2 - R_1 - s_1)$

Pareja 6 $(P_{1,1 \oplus 2,1}, P_{2,2}) = (e_1 - \boldsymbol{R_1} - s_1,\ e_2 - R_4 - R_3 - s_2)$

(NOTA: Se ha marcado en negrita el nodo de cruce)

Las parejas 4 y 5 no son disjuntas, por lo que se descartan; en cambio las parejas 3 y 6 sí lo son, y ambas con el mismo coste, por tanto, la solución del algoritmo es, por ejemplo, la pareja 6.

Solución:

- $P^s_{1,1} = e_1 - R_1 - s_1$
- $P^s_{2,2} = e_2 - R_4 - R_3 - s_2$

Nótese que el ejemplo propuesto es un caso extremo en el que los caminos ofrecidos por e_1 y e_2 se cruzan todos con todos. Es un ejemplo del caso peor de cruces entre caminos, pero es un buen ejemplo de funcionamiento del algoritmo propuesto.

Como puede verse la solución obtenida no es óptima, puesto que pasar simplemente por R_1 para $P^s_{1,1}$ y R_4 para $P^s_{2,2}$ tiene menor coste. Si bien, la solución computada utiliza los caminos computados por e_1 y e_2, respetando por tanto el cómputo de estos . Esta solución es un compromiso entre obtener una solución, minimizar el coste y respetar los caminos originales computados por e_1 y e_2. Recordemos que los caminos de los que se parte en este ejemplo no son los óptimos, pero esto puede ser por cumplir requisitos de QoS o criterios locales.

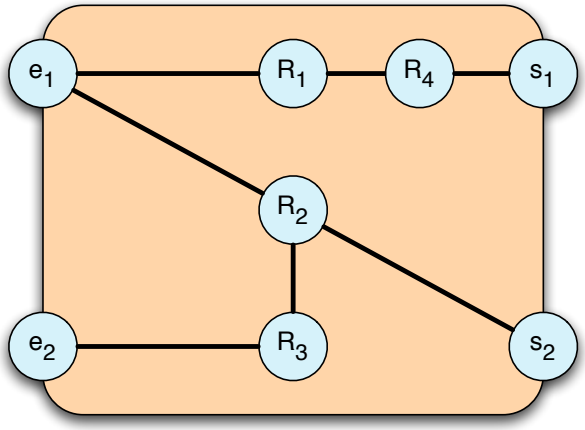


Figura 4.8: Ejemplo 3 de utilización del algoritmo. Topología trampa.

En la demostración se probará que cumplidas las condiciones de partida siempre existe al menos una solución y que el algoritmo propuesto siempre es capaz de encontrar una de ellas, además esta la obtiene con un coste computacional pequeño. Sin embargo, también es posible reconstruir un grafo parcial con los caminos obtenidos y ejecutar sobre él un algoritmo de cálculo de caminos disjuntos como el de Suurballe [9] y obtener así la pareja óptima sin tener en cuenta las decisiones locales.

Ejemplo 3 En este último ejemplo se muestra cómo actúa el algoritmo ante una de las topologías trampa que se estudiaron en la sección 2.4.4. En la figura 4.8 se encuentra redibujada en un esquema de apariencia similar al resto de ejemplos la topología trampa que se mostró en la figura 2.6.

Como se puede ver, esta red no es una red 2-conectada y además desde e_2 no es posible encontrar un par de caminos disjuntos a los nodos de salida. En este caso e_2 computará los caminos "lo más disjuntos posible"[5]. En la demostración del algoritmo (sección 4.4.2) también se demostrará que para poder encontrar una solución no siempre es necesario que $P_{1,1}$ y $P_{1,2}$ sean completamente disjuntos y que $P_{2,1}$ y $P_{2,2}$ también lo sean[6].

Con lo que tenemos que:
$P_{1,1} = e_1 - R_1 - R_4 - s_1$
$P_{1,2} = e_1 - R_2 - s_2$
$P_{2,1} = e_2 - R_3 - R_2 - e_1 - R_1 - R_4 - s_1$
$P_{2,2} = e_2 - R_3 - R_2 - s_2$

[5]Tal y como se indicó en el algoritmo
[6]Piense en el caso en el que la solución es directamente $P_{1,1}$ y $P_{2,2}$.

Ejecutando el algoritmo y empezando por el paso 1 tenemos las parejas:

Pareja 1 $(P_{1,1}, P_{2,2}) = (e_1 - R_1 - R_4 - s_1,\ e_2 - R_3 - R_2 - s_2)$

Pareja 2 $(P_{1,2}, P_{2,1}) = (e_1 - R_2 - s_2,\ e_2 - R_3 - R_2 - e_1 - R_1 - R_4 - s_1)$

La pareja 2 no es disjunta, con lo que esta queda descartada. Sólo queda la pareja 1 que sí lo es, con lo que la solución del algoritmo es la pareja 1.

Solución:

- $P^s_{1,1} = e_1 - R_1 - R_4 - s_1$
- $P^s_{2,2} = e_2 - R_3 - R_2 - s_2$

Nótese que se ha evitado la trampa sin caer en ella en primera instancia.

La siguiente sección está dedicada a la demostración de que, cumplidas las condiciones de partida, el algoritmo APD siempre encuentra una solución, y que por tanto dadas las condiciones de partida siempre existe una solución.

4.4.2. Demostración

En este apartado se pretende demostrar el teorema 4.1 y el correcto funcionamiento del algoritmo APD propuesto.

La demostración se basa en los caminos propuestos en la figura 4.3. Se verá que todas las posibles situaciones de cruces de caminos que pueden producirse se solucionan con alguna de las parejas de caminos propuestas en el algoritmo APD (ecuaciones 4.1 - 4.6). De este modo quedará demostrado que con los datos de partida, $P_{1,1}$, $P_{1,2}$, $P_{2,1}$ y $P_{2,2}$ existe siempre al menos una solución y que el algoritmo es capaz de calcularla.

El árbol de los posibles situaciones que se pueden producir, desde el punto de vista de los cruces entre los caminos, se muestra en la tabla 4.1.

En esta tabla están contenidos todos los posibles cruces entre los cuatro caminos. Para todos los casos existe una solución válida, es decir que los diferentes $P^s_{\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}}$ existen y son disjuntos siempre que $P_{1,1} \perp P_{1,2}$ y $P_{2,1} \perp P_{2,2}$.

$$
\begin{cases}
P_{1,1} \perp P_{2,2} & (1) \\
P_{1,1} \not\perp P_{2,2} \begin{cases}
P_{1,2} \perp P_{2,1} & (2) \\
P_{1,2} \not\perp P_{2,1} \begin{cases}
P_{1,1} \perp P_{2,1},\ P_{1,2} \perp P_{2,2} & (3) \\
P_{1,1} \perp P_{2,1},\ P_{1,2} \not\perp P_{2,2} & (4) \\
P_{1,1} \not\perp P_{2,1},\ P_{1,2} \perp P_{22} & (5) \\
P_{1,1} \not\perp P_{2,1}, P_{1,2} \not\perp P_{2,2} & (6)
\end{cases}
\end{cases}
\end{cases}
\tag{4.7}
$$

Tabla 4.1: Árbol de posibles situaciones desde el punto de vista de los cruces entre los caminos.

Las situaciones (1) y (2) de (4.7) son triviales. Si $P_{1,1}$ y $P_{2,2}$ son disjuntos estos mismos caminos son solución y, por tanto, $\langle \boldsymbol{P_1},\ \boldsymbol{P_2} \rangle^{\boldsymbol{s}} = \langle \boldsymbol{P_{1,1}},\ \boldsymbol{P_{2,2}} \rangle$. De igual modo cuando $P_{1,2}$ y $P_{2,1}$ son disjuntos, estos dos caminos son solución y, por tanto, $\langle \boldsymbol{P_1},\ \boldsymbol{P_2} \rangle^{\boldsymbol{s}} = \langle \boldsymbol{P_{1,2}},\ \boldsymbol{P_{2,1}} \rangle$. Estas dos posibles soluciones son las que se han propuesto como paso 1 del algoritmo.

En las situaciones restantes, de la (3) a la (6) de la tabla 4.1, ocurre que $P_{1,1} \not\perp P_{2,2}$ y $P_{1,2} \not\perp P_{2,1}$. Para obtener las soluciones de estas situaciones se estudiarán los siguientes cuatro casos.

Caso A $P_{1,1} \perp P_{2,1}$, esto es $P_{1,1}$ y $P_{2,1}$ no se cruzan. El **caso A** es válido como solución para las situaciones (3) y (4) de la tabla 4.1. Uno de los dos caminos solución puede ser $P^s_{2,1} = P_{2,1}$, para lo que hay que encontrar el disjunto que vaya desde el nodo e_1 de entrada al nodo s_2 de salida, es decir, $P^s_{1,2}$. Como $P_{1,1}$ es disjunto a $P_{2,1}$ (condición de partida del **Caso A**), se puede utilizar este camino como salida desde el nodo e_1. Hay que llegar al nodo s_2 y dado que $P_{1,1} \not\perp P_{2,2}$ (condición de partida) se puede continuar por $P_{2,2}$ desde el cruce hasta el nodo de salida s_2. Debido a que $P_{2,1}$ y $P_{2,2}$ son disjuntos se ha encontrado un $P^s_{1,2} = P_{1,1 \oplus 2,2}$ disjunto a $P^s_{2,1}$. La solución del **Caso A** queda por tanto $\langle \boldsymbol{P_1},\ \boldsymbol{P_2} \rangle^{\boldsymbol{s}} = \langle [\boldsymbol{P_{1,1 \oplus 2,2}}],\ \boldsymbol{P_{2,1}} \rangle$.

Caso B $P_{1,2} \perp P_{2,2}$, esto es $P_{1,2}$ y $P_{2,2}$ son los disjuntos, no se cruzan. Es el caso homólogo al **Caso A**. Este caso cumple las condiciones de las situaciones (3) y (5) de la tabla 4.1. El razonamiento es similar al anterior; como primer camino solución se puede tomar el camino directo $P^s_{1,2} = P_{1,2}$. Ahora hay que computar

$P^s_{2,1}$, camino que sale de e_2 y llega a s_1. El camino disjunto a $P_{1,2}$ que sale de e_2 es $P_{2,2}$ (condición de partida del **Caso B**). Cuando este se cruza con $P_{1,1}$ se cambia de camino para llegar a s_1. Es condición de partida que $P_{1,2}$ y $P_{1,1}$ son disjuntos. con lo que $P^s_{2,1} = P_{2,2 \oplus 1,1}$ disjunto a $P^s_{1,2}$. La solución del **Caso B** queda por tanto $\langle \boldsymbol{P_1}, \boldsymbol{P_2} \rangle^{\boldsymbol{s}} = \langle \boldsymbol{P_{1,2}}, [\boldsymbol{P_{2,2 \oplus 1,1}}] \rangle$.

Con los **Casos A** y **B** se ha dado solución a las situaciones (3), (4) y (5) de la tabla 4.1. Sólo queda encontrar una solución para la última de las situaciones, la (6). En esta opción todos los caminos de partida se cruzan con todos, $P_{11} \not\perp P_{21}$, $P_{11} \not\perp P_{22}$, $P_{12} \not\perp P_{21}$ y $P_{12} \not\perp P_{22}$, y es importante tener en cuenta la posición relativa de los cruces. Si se parte teniendo en cuenta el cruce $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ existen 4 opciones, que este cruce sea anterior o posterior a los otros cruces de $P_{1,1}$ y $P_{2,2}$. A continuación se analizan cada una de estas opciones para a continuación analizar los casos que se derivan de ellas:

1. $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ es anterior a $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$: se corresponde con el **Caso A**, en el que $P_{2,1}$ es disjunto a $P_{1,1}$ hasta $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ (que es anterior a $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$) y $P_{2,1}$ es disjunto a $P_{2,2}$.

2. $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ es anterior a $\overset{P_{1,2}}{\underset{P_{2,2}}{\oplus}}$: se corresponde con el **Caso B**, en el que $P_{1,2}$ es disjunto a $P_{2,2}$ hasta $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ (que es anterior a $\overset{P_{1,2}}{\underset{P_{2,2}}{\oplus}}$) y $P_{1,2}$ es disjunto a $P_{1,1}$.

3. $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ es posterior a $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$: se corresponde con un nuevo caso a estudiar, el **Caso C**. Siguiendo un razonamiento similar a los casos anteriores se tiene que $P_{2,2}$ es disjunto a $P_{1,1}$ hasta $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$ (que es anterior a $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$) y $P_{2,2}$ es disjunto a $P_{2,1}$. Se estudiará a continuación.

4. $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ es posterior a $\overset{P_{1,2}}{\underset{P_{2,2}}{\oplus}}$: se corresponde con otro nuevo caso a estudiar, el **Caso D**, que en un razonamiento similar se concluye que $P_{2,2}$ es disjunto a $P_{1,1}$ hasta $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$ (que es anterior a $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$) y $P_{2,2}$ es disjunto a $P_{2,1}$. También se estudiará a continuación.

Nótese que en ninguna opción los cruces que se suponen anteriores o posteriores pueden darse en el mismo punto, puesto que se estarían cruzando en un mismo

punto, por ejemplo para 1.) $P_{1,1}$, $P_{2,1}$ y $P_{2,2}$, dos caminos que son disjuntos como condición inicial, en el ejemplo, $P_{2,1}$ y $P_{2,2}$. Por lo tanto, estos cruces nunca podrán ser simultáneos y siempre se producirá una de las cuatro situaciones analizadas.

La situación (6) queda resuelta por alguno de los **Casos A**, **B**, **C** o **D**. Se han detallado los dos primeros, a continuación se detallarán los dos últimos.

Caso C Todos los caminos se cruzan y además $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ es posterior a $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$. Como primer camino solución puede tomarse $P^s_{2,2} = P_{2,2}$. Hay que encontrar un $P^s_{1,1}$ disjunto a este, es decir, que vaya desde el nodo de entrada e_1 al nodo de salida s_1. Camino disjunto a $P_{2,2}$ que salga de e_1, al menos hasta que se cruza con él, es $P_{1,1}$. Para evitar el cruce con $P_{2,2}$ y puesto que cruza con $P_{2,1}$ con anterioridad (condición de partida del **Caso C**) el camino cambia en el cruce al camino $P_{2,1}$, disjunto a $P_{2,2}$ y que además llega a s_1. Queda, por tanto, como segundo camino solución $P^s_{1,1} = P_{1,1 \oplus 2,1}$ disjunto a $P^s_{2,2}$. Y la solución del **Caso C** queda por tanto $\langle \boldsymbol{P_1}, \boldsymbol{P_2} \rangle^{\boldsymbol{s}} = \langle [\boldsymbol{P_{1,1 \oplus 2,1}}], \boldsymbol{P_{2,2}} \rangle$.

Caso D Todos los caminos se cruzan y $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ es posterior a $\overset{P_{1,2}}{\underset{P_{2,2}}{\oplus}}$. Es el Caso hómologo al **Caso C**. Como primer camino solución puede tomarse, en este caso, $P^s_{1,1} = P_{1,1}$. Por tanto, hay que encontrar un $P^s_{2,2}$, camino entre el nodo de entrada e_2 al nodo de salida s_2, disjunto a este. En este caso, camino disjunto a $P_{1,1}$ que salga de e_2, hasta que se crucen, es $P_{2,2}$. Para evitar el cruce con $P_{1,1}$ y puesto que cruza con $P_{1,2}$ con anterioridad (condición de partida del **Caso D**) se cambia al camino $P_{1,2}$ en el cruce, disjunto a $P_{1,1}$ y que además llega a s_2. Queda como segundo camino solución $P^s_{2,2} = P_{2,2 \oplus 1,2}$ disjunto a $P^s_{1,1}$. Y la solución del **Caso C** queda por tanto $\langle \boldsymbol{P_1}, \boldsymbol{P_2} \rangle^{\boldsymbol{s}} = \langle \boldsymbol{P_{1,1}}, [\boldsymbol{P_{2,2 \oplus 1,2}}] \rangle$.

Las parejas solución de estos 4 casos, que resuelven las situaciones (3), (4), (5) y (6), son las propuestas en el segundo paso del algoritmo. En la Tabla 4.2 se resumen las diferentes soluciones para todas las situaciones posibles.

4.4.3. Análisis del algoritmo

En la demostración planteada anteriormente realmente se han demostrado dos cosas; primera, siempre que e_1 y e_2 puedan calcular de manera independiente dos caminos disjuntos a los nodos de salida, s_1 y s_2, existen al menos un par de caminos disjuntos que unen estos nodos de entrada y salida; y segundo, el algoritmo planteado es capaz de calcular una de estas soluciones en todos los casos.

$$
\begin{cases}
P_{1,1} \perp P_{2,2} \;\Rightarrow\; \langle \boldsymbol{P_1},\, \boldsymbol{P_2}\rangle^{\boldsymbol{s}} = \left\langle \boldsymbol{P^s_{1,1}},\, \boldsymbol{P^s_{2,2}} \right\rangle = \langle \boldsymbol{P_{1,1}},\, \boldsymbol{P_{2,2}}\rangle & (1) \\
P_{1,1} \not\perp P_{2,2} \begin{cases}
P_{1,2} \perp P_{2,1} \;\Rightarrow\; \langle \boldsymbol{P_1},\, \boldsymbol{P_2}\rangle^{\boldsymbol{s}} = \left\langle \boldsymbol{P^s_{1,2}},\, \boldsymbol{P^s_{2,1}} \right\rangle = \langle \boldsymbol{P_{1,2}},\, \boldsymbol{P_{2,1}}\rangle & (2) \\
P_{1,2} \not\perp P_{2,1} \begin{cases}
P_{1,1} \perp P_{2,1}, P_{1,2} \perp P_{2,2} \;\Rightarrow\; \textbf{Caso A y Caso B} & (3) \\
P_{1,1} \perp P_{2,1}, P_{1,2} \not\perp P_{2,2} \;\Rightarrow\; \textbf{Caso A} & (4) \\
P_{1,1} \not\perp P_{2,1}, P_{1,2} \perp P_{2,2} \;\Rightarrow\; \textbf{Caso B} & (5) \\
P_{1,1} \not\perp P_{2,1}, P_{1,2} \not\perp P_{2,2} \;\Rightarrow\; \textbf{Caso A}, \textbf{Caso B}, \textbf{Caso C y Caso D} & (6)
\end{cases}
\end{cases}
\end{cases}
\tag{4.8}
$$

Donde,

$$\text{Caso A}: \quad \langle \boldsymbol{P_1},\, \boldsymbol{P_2}\rangle^{\boldsymbol{s}} = \left\langle \boldsymbol{P^s_{1,2}},\, \boldsymbol{P^s_{2,1}} \right\rangle = \langle [\boldsymbol{P_{1,1} \oplus 2,2}],\, \boldsymbol{P_{2,1}}\rangle \tag{4.9}$$

$$\text{Caso B}: \quad \langle \boldsymbol{P_1},\, \boldsymbol{P_2}\rangle^{\boldsymbol{s}} = \left\langle \boldsymbol{P^s_{1,2}},\, \boldsymbol{P^s_{2,1}} \right\rangle = \langle \boldsymbol{P_{1,2}},\, [\boldsymbol{P_{2,2} \oplus 1,1}]\rangle \tag{4.10}$$

$$\text{Caso C}: \quad \langle \boldsymbol{P_1},\, \boldsymbol{P_2}\rangle^{\boldsymbol{s}} = \left\langle \boldsymbol{P^s_{1,1}},\, \boldsymbol{P^s_{2,2}} \right\rangle = \langle [\boldsymbol{P_{1,1} \oplus 2,1}],\, \boldsymbol{P_{2,2}}\rangle \tag{4.11}$$

$$\text{Caso D}: \quad \langle \boldsymbol{P_1},\, \boldsymbol{P_2}\rangle^{\boldsymbol{s}} = \left\langle \boldsymbol{P^s_{1,1}},\, \boldsymbol{P^s_{2,2}} \right\rangle = \langle \boldsymbol{P_{1,1}},\, [\boldsymbol{P_{2,2} \oplus 1,2}]\rangle \tag{4.12}$$

Tabla 4.2: Árbol de posibles situaciones desde el punto de vista de los cruces entre los caminos con sus soluciones.

Es fácil ver que existen ocasiones en las que hay más de una posible solución. Por ejemplo, en algún caso la pareja $P_{1,1 \oplus 2,1}$ y $P_{2,2 \oplus 1,2}$ podría ser una solución válida, pero no aporta nada a la demostración. Nótese que todas las soluciones propuestas por el algoritmo tienen al menos uno de los caminos calculados por los nodos de entrada. De este modo queda un algoritmo sencillo con solución siempre y que tiene en cuenta las políticas locales de decisión.

Como ya se ha indicado, en el caso de querer prescindir de las políticas locales y buscar la solución óptima es posible construir un subgrafo con los caminos $P_{1,1}$, $P_{1,2}$, $P_{2,1}$ y $P_{2,2}$ y aplicar el algoritmo de Suurballe [9] para obtener la solución óptima.

Una de las suposiciones que se ha realizado es que la red sobre la que se está trabajando es 2-conectada. A continuación se analizará qué sucede si esta suposición no es cierta y por tanto no es posible computar caminos disjuntos desde cada nodo

de entrada a los nodos de salida. O que aún siendo una red 2-conectada la existencia de requisitos de QoS hace que estos caminos no se puedan computar por falta de recursos.

4.4.3.1. Falta un camino

Supongamos que uno de los caminos no se ha podido calcular, por ejemplo el $P_{1,2}$[7], o bien no es completamente disjunto con $P_{1,1}$. En este caso las situaciones se simplifican, quedan reflejadas en la tabla 4.3.

$$\begin{cases} P_{1,1} \perp P_{2,2} & (1) \\ P_{1,1} \not\perp P_{2,2} \begin{cases} P_{1,1} \perp P_{2,1} & (2) \\ P_{1,1} \not\perp P_{2,1} & (3) \end{cases} \end{cases} \tag{4.13}$$

Tabla 4.3: Árbol de casos posibles desde el punto de vista de los cruces entre los caminos cuando falta $P_{1,2}$.

Siguiendo un razonamiento paralelo al realizado durante la demostración del algoritmo podemos concluir lo siguiente. La situación (1) sigue siendo trivial, con la misma solución $\langle P_1,\, P_2\rangle^s = \langle P_{1,1},\, P_{2,2}\rangle$. En la situación (2) la solución se corresponde con el **Caso A**, $\langle P_1,\, P_2\rangle^s = \langle [P_{1,1} \oplus {}_{2,2}],\, P_{2,1}\rangle$. La situación (3) tiene dos posibilidades, que $\overset{P_{1,1}}{\underset{P_{2,2}}{\oplus}}$ sea anterior o posterior a $\overset{P_{1,1}}{\underset{P_{2,1}}{\oplus}}$. Si el cruce es anterior, la solución previa, el **Caso A**, es válida. En el caso de que sea posterior, entonces la solución se corresponde con el **Caso C**, $\langle P_1,\, P_2\rangle^s = \langle [P_{1,1} \oplus {}_{2,1}],\, P_{2,2}\rangle$.

Las tres soluciones requeridas son parejas propuestas en el algoritmo, por tanto, no hay que preocuparse a la hora de utilizar el algoritmo de si algún camino falta o no es disjunto con su pareja, porque aún así existe siempre una solución y se puede calcular con el algoritmo propuesto. Tenemos pues el siguiente corolario al teorema 4.1.

Corolario 4.1.1. *Con tres de los cuatro caminos disjuntos siempre es posible obtener una solución.*

[7] Para el resto de caminos se puede seguir un razonamiento similar o renombrar los nodos de entrada y salida para hacer coincidir el camino ausente con el $P_{1,2}$.

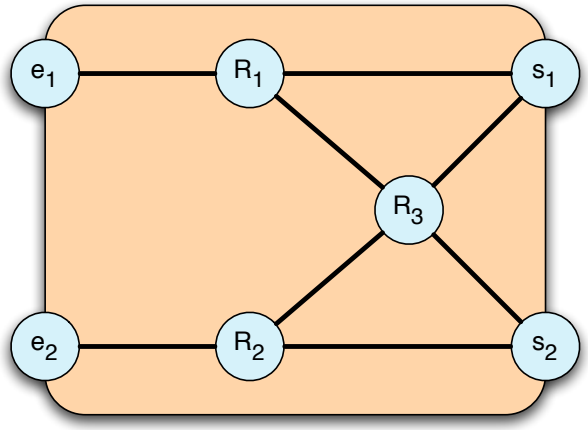


Figura 4.9: Ejemplo 4 de utilización del algoritmo. Caminos de partida no completamente disjuntos.

4.4.3.2. No hay caminos disjuntos desde cada nodo de entrada

Estudiemos un caso peor, ninguna de las dos parejas es disjunta. Aún así, en algunos casos, todavía se puede encontrar una solución. Si uno de los caminos desde una de las entradas no existe o comparte gran parte del camino con su par (e.g. $P_{1,1}$ y $P_{1,2}$), entonces, los caminos del otro nodo ($P_{2,1}$ y $P_{2,2}$ en este ejemplo) deben ser disjuntos a partir del primer cruce con $P_{1,1}$ o $P_{1,2}$, dependiendo de cuál se esté utilizando como parte de la solución. Es por esto que los caminos en cuestión pueden compartir recorrido hasta que empiecen a cruzarse. Por lo que realmente los caminos calculados por e_1 y e_2 deben ser lo más disjuntos posible, entendiendo por "lo más disjuntos posible" que en caso de que no puedan ser completamente disjuntos es mejor que sea el trayecto final de los caminos el que sea disjunto. Sirva como ejemplo el propuesto en la figura 4.9.

Para la que tenemos:
$P_{1,1} = e_1 - R_1 - s_1$
$P_{1,2} = e_1 - R_1 - R_3 - s_2$
$P_{2,1} = e_2 - R_2 - R_3 - s_1$
$P_{2,2} = e_2 - R_2 - s_2$

Ejecutando el algoritmo y empezando por el paso 1 se tienen las parejas:

Pareja 1 $(P_{1,1}, P_{2,2}) = (e_1 - R_1 - s_1,\ e_2 - R_2 - s_2)$

Pareja 2 $(P_{1,2}, P_{2,1}) = (e_1 - R_1 - R_3 - s_2,\ e_2 - R_2 - R_3 - s_1)$

La pareja 2 no es disjunta, con lo que esta se descarta. Sólo queda la pareja 1 que

sí lo es, con lo que la solución del algoritmo es la pareja 1.

Solución:

- $P^s_{1,1} = e_1 - R_1 - s_1$
- $P^s_{2,2} = e_2 - R_2 - s_2$

Se ha visto que si la red es 2-conectada siempre es posible encontrar los $P_{i,j}$ y que a partir de estos, se ha demostrado que, siempre es posible encontrar un par de caminos disjuntos. Pero qué ocurre si existe una solución, y la red no es 2-conectada, ¿es posible computar siempre esta solución utilizando el algoritmo descrito?.

Ya se ha visto que aún en el caso de que uno de los cuatro caminos no esté presente siempre existe una solución. Entonces hay que analizar casos más restrictivos que este, partiendo de que al menos hay un par de caminos disjuntos entre los nodos de entrada y salida, es decir, existe al menos una solución válida. Para ello se analizarán estos casos y se verá que siempre se obtienen los caminos suficientes para llegar a la solución.

El peor caso es cuando sólo hay una solución posible al problema, es decir sólo hay un par de caminos disjuntos $P^s_{\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}\left\{\begin{smallmatrix}1\\2\end{smallmatrix}\right\}}$. Para los ejemplos se va a utilizar $P^s_{1,1}$ y $P^s_{2,2}$, pero el razonamiento sería similar para $P^s_{1,1}$ y $P^s_{2,2}$. Hay que asegurar que los nodos y enlaces que forman estos caminos forman parte de los los $P_{i,j}$. Existen tres posibles situaciones, que $P^s_{1,1}$ y $P^s_{2,2}$ sean los únicos caminos que unan los nodos de entrada y salida, que existan varios y no se elijan estos por e_1 y e_2 o que existan varios y sí se elijan.

El primer caso es trivial, si son los únicos caminos e_1 calculará como camino a s_1 el único existente, $P_{1,1} = P^s_{1,1}$, lo mismo ocurrirá con $P_{2,2} = P^s_{1,1}$, siendo indistintos los otros caminos calculados, $P_{1,2}$ y $P_{2,1}$, puesto que $P_{1,1}$ y $P_{2,2}$ son solución.

En el caso de que existan más opciones que unan los nodos de entrada con los de salida pero los nodos de entrada no elijan los caminos solución y los elegidos no sirvan para tal fin es porque $P_{1,1}$ y $P_{2,2}$ se cortan en algún punto. Este punto de corte asegura que hay un posible camino entre e_1 y s_2 y entre e_2 y s_1, con lo que existirían y los nodos de entrada podrían calcular los caminos $P_{1,2}$ y $P_{2,1}$. Si estos caminos siguen a sus homólogos hasta el cruce y a partir de ahí continúan por el otro camino lo que ocurre es que, por ejemplo, $P_{1,2}$ coincide con $P_{1,1}$ hasta su cruce con $P_{2,2}$ y a partir del cruce continúa por $P_{2,2}$ hasta s_2. En esta situación en realidad la única información que se tiene son los caminos $P_{1,1}$ y $P_{2,2}$, puesto que los otros dos son coincidentes en todo momento con parte de estos. Pero esta situación no puede darse porque e_1 y e_2 deben calcular sus caminos lo más disjuntos posibles y en este caso existen dos caminos más disjuntos, puesto que hay una alternativa más (el camino

solución) para llegar a los nodos de salida. De este modo $P_{1,1}$ y $P_{1,2}$ no serían tan coincidentes y el algoritmo APD obtendría la solución.

Veamos un ejemplo. En la figura 4.10 se muestra esta situación; la única solución posible sería $P^*_{1,1} = e_1 - R_1 - R_4 - s_1$ y $P^*_{2,2} = e_2 - R_2 - R_5 - s_2$. La elección de caminos que podría elegir e_1 que no funcionaría sería $P_{1,1} = e_1 - R_1 - R_3 - R_4 - s_1$ y $P_{1,2} = e_1 - R_1 - R_3 - R_5 - s_2$ si además e_2 elige $P_{2,1} = e_2 - R_2 - R_3 - R_4 - s_1$ y $P_{2,2} = e_2 - R_2 - R_3 - R_5 - s_2$. Pero e_1, y e_2, no elegirían este par de caminos puesto que hay otra pareja más disjunta, $P_{1,1} = e_1 - R_1 - R_4 - s_1$ y $P_{1,2} = e_1 - R_1 - R_3 - R_5 - s_2$, en la que sólo coinciden en un nodo. Con esta elección sí que es posible encontrar la solución.

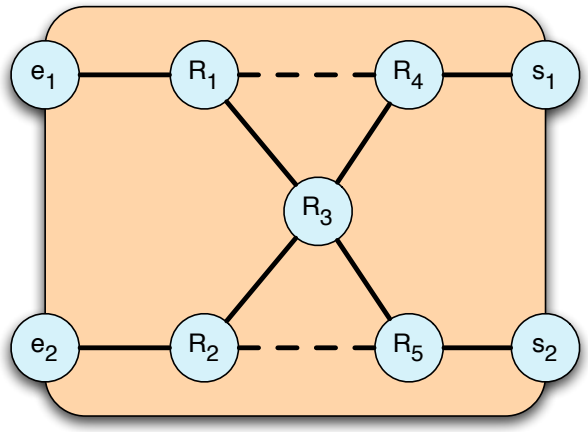


Figura 4.10: Solución única.

Por tanto, si se mantiene la máxima de que los caminos calculados sean lo más disjuntos posible siempre que exista una solución el algoritmo APD será capaz de obtenerla.

Definición 4.3. *Dos caminos que parten de un mismo origen son lo más* disjuntos posibles *si en el caso de no poder ser completamente disjuntos si la parte de camino común es lo más pequeña posible, compacta y situada en el inicio de los caminos.*

4.5. Generalización

En este apartado se va a realizar un breve estudio del caso general en el que se desea calcular N caminos disjuntos entre los nodos de entrada, e_i, y los nodos de salida, s_j. Hay N_e nodos de entrada y N_s nodos de salida, y todos los nodos de entrada

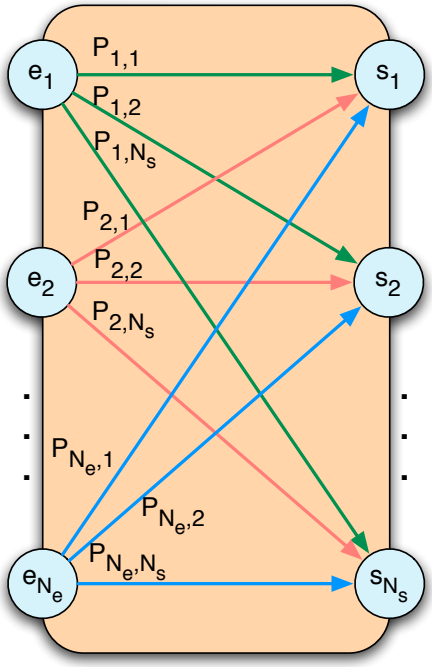


Figura 4.11: Caminos disjuntos desde cada nodo de entrada a los nodos de salida. Caso general

han calculado los N_s caminos disjuntos, uno a cada nodo de salida, necesarios. En la figura 4.11 se muestra un esquema de la situación de partida.

Bajo estas condiciones existen tres posibles situaciones;

$N_e = N_s$ Si el número de nodos de entrada y salida son el mismo, entonces el número de caminos disjuntos, N, que se pueden trazar es $N = N_e = N_s$.

$N_e < N_s$ Si el número de nodos de entrada es menor que el número de nodos de salida entonces sólo se podrán calcular $N = N_e$ caminos disjuntos, puesto que estos ocuparán cada uno un nodo de entrada, cualquier camino de más necesitaría repetir nodo de entrada y dejaría de ser disjunto.

$N_e > N_s$ Si el número de nodos de entrada es mayor que el número de nodos de salida entonces sólo se podrán calcular $N = N_s$ caminos disjuntos, puesto que estos ocuparán cada uno un nodo de salida, cualquier camino de más necesitaría repetir nodo de salida y dejaría de ser disjunto.

En los dos últimos casos se puede realizar una preselección de qué nodos de entrada y salida se quieren interconectar.

Aquí sólo se tratará de mostrar que es posible obtener los caminos disjuntos a partir de la información de partida. No se va a realizar una demostración exhaustiva puesto que este caso no tiene especial interés en la realización de esta tesis doctoral. Posibles aplicaciones para este tipo de cálculo podría ser realizar reparto de carga en conexiones interdominio entre distintos caminos. Al igual que en el caso de 2 nodos los caminos de partida calculados por los nodos de entrada deben ser lo más disjuntos posible, pudiendo alguno de ellos ni siquiera haber sido calculado.

Se deben encontrar N caminos disjuntos de entre los N^2 caminos computados por los nodos de entrada, estos caminos pueden cruzarse entre sí. Sólo se analizarán dos posibles casos, aquel en el que ningún camino se cruza con ningún otro camino, y aquel en el que todos los caminos se cruzan con todos.

En el primer caso, es posible elegir directamente los $P_{i,i}$ directamente como solución, puesto que al no cruzarse con ninguno son disjuntos con todos.

En el segundo caso, todos los caminos se cruzan con todos, esta situación se corresponde con una red tipo *crossbar*. En un *crossbar* siempre es posible conectar cualquier entrada con cualquier salida sin necesidad de compartir los caminos [84].

4.6. Conclusiones

En este capítulo se ha presentado un algoritmo para computar dos caminos disjuntos dentro de un dominio, desde dos nodos de entrada a dos nodos de salida. Para poder realizar estos cálculos el algoritmo se basa en los caminos computados de manera independiente por cada uno de los nodos de entrada al dominio, lo que permitiría poder computar caminos disjuntos dentro del dominio sin necesidad de compartir toda la información topológica de la red.

El algoritmo de cómputo propuesto respeta al máximo las decisiones locales de los nodos de entrada, intentado ajustarse al máximo a los caminos propuestos por los nodos de entrada.

Se ha visto cómo utilizando esta metodología es posible computar un par de caminos disjuntos siempre que estos existan, es decir, si hay una solución el algoritmo es capaz de computarla sin caer en ninguna trampa y, por tanto, sin necesidad de utilizar mecanismos de *backtracking*.

Se ha generalizado la utilización del algoritmo al cómputo de N caminos disjuntos, aunque esta aplicación no tiene interés en el marco de esta Tesis Doctoral.

La metodología propuesta está indicada para ser utilizada dentro de un dominio. En capítulos siguientes se verá cómo utilizarla para resolver problemas multidominio.

Capítulo 5

Cómputo de caminos disjuntos por áreas IGP

En este capítulo se va a explicar cómo utilizar las propiedades del algoritmo expuesto en el capítulo 4 en un mecanismo distribuido de cómputo de caminos disjuntos extremo a extremo. Para ello primero se explicará dicho mecanismo propuesto mediante un ejemplo. A continuación, se procederá a explicar cómo utilizar este mecanismo incluso cuando los dominios estén divididos en diferentes áreas OSPF, lo que implica sólo un conocimiento parcial de la red interior. Para esta tarea también se utilizará el algoritmo del capítulo 4. Por último, se incluye un resumen junto a las conclusiones finales de este capítulo.

5.1. Introducción

Ya se ha visto al hablar sobre los mecanismos PPRO y ARO[1] que el problema de éstos es la posibilidad de caer en trampas topológicas. También se hizo hincapié en que utilizando PCE esto podía ser evitado computando todos los posibles caminos en cada dominio para elegir de cada dominio aquellos que resulten en los caminos extremo a extremo óptimos, sin embargo con algunos problemas y carencias, comentadas en 2.4 y 3.

Si no se dispone de un PCE en cada dominio entonces es necesario utilizar un mecanismo distribuido. La característica común de los mecanismos distribuidos vistos es que el LSR de entrada de un dominio es el que computa el tramo del LSP dentro de su dominio. Este hecho fuerza el punto de entrada al siguiente dominio, lo que permite la caída en trampas topológicas que obligan a utilizar mecanismos de *cranckback* que dificultan y retardan el establecimiento de los caminos. existen dos

[1] Sección 2.4.

formas de evitar este tipo de problematica; Primera, que el LSR de entrada compute todos los posibles pares de caminos disjuntos dentro del dominio, desde cada posible par de LSRs de entrada a cada posible par de LSRs de salida. Segunda, simplificar el método anterior de modo que sea más escalable y que a la vez no caiga en trampas topológicas.

Para poder utilizar la primera de ellas es necesario que el LSR de entrada tenga conocimiento del grafo completo de toda la red. Si el dominio, Sistema Autónomo, está dividido en áreas IGP, entonces esto no ocurre, el LSR de entrada no tiene el conocimiento suficiente de la red como para poder computar todos esos caminos disjuntos, de hecho es posible que no tenga conocimiento ni para calcular un par de ellos y tenga que recurrir a un mecanismo de cómputo interdominio.

5.2. Mecanismo de cómputo distribuido y establecimiento de caminos disjuntos utilizando el algoritmo propuesto en el capítulo 4

En esta sección se propone un posible mecanismo de descubrimiento y señalización de dos caminos disjuntos interdominio utilizando las características del algoritmo detallado en el capítulo 4. Es posible proponer diferentes mecanismos que exploten las características del algoritmo; aquí presentaremos el que, en nuestra opinión, es el más adecuado para el caso interdominio cuando los dominios son Sistemas Autónomos.

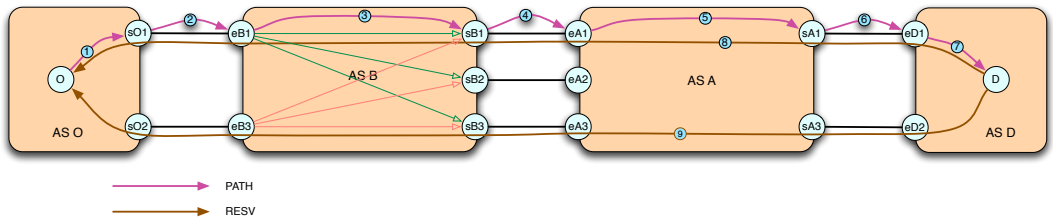


Figura 5.1: Esquema de cómputo distribuido de caminos disjuntos utilizando el algoritmo propuesto en el capítulo 4.

Para ejemplificar el mecanismo general se utilizará la red de la figura 5.1, como puede verse se indican los mensajes de PATH y RESV, con un número se indica la secuencia de los mensajes. El LSR *O*, del dominio *AS O*, quiere computar y establecer dos caminos interdominio disjuntos, un camino principal y otro de respaldo, con el LSR *D*, del dominio *AS D*. Como puede verse en la figura esto implica pasar por los dominios *AS A* y *AS B*.

Ya se ha visto que utilizando los mecanismos de PPRO y ARO, sección 2.4, es posible caer en una trampa topológica. Gracias al algoritmo del capítulo 4 y al mecanismo distribuido de cómputo explicado a continuación es posible calcular estos caminos sin caer en una trampa topológica.

El LSR *O* computa dos caminos disjuntos, uno a cada uno de los nodos de entrada del siguiente AS, el que le permite alcanzar *D*: $eB1$ y $eB3$[2]. Una vez computados envía un mensaje de PATH a uno de ellos, el elegido para el camino principal, e.g. $eB1$, con los caminos computados. Estos caminos se incluyen en un nuevo objeto RSVP-TE, el DPMMO (*Disjoint Paths Multiples Origins Multiples Destinations Object*), explicado en el anexo B.2.

Cuando $eB1$ recibe el mensaje de PATH este obtiene (computa o recibe) de cada LSR de entrada del dominio (sólo aquellos que además comparten dominio "previo", *AS 0*, con $eB1$) los caminos disjuntos hasta los LSRs de entrada del dominio siguiente, *ASA*[3]. Estos caminos se corresponden con $eB1-eA1$, $eB1-eA2$ y $eB1-eA3$ por un lado y $eB3-eA1$, $eB3-eA2$ y $eB3-eA3$ por otro. Cada uno de los grupos son disjuntos entre sí.

Existen dos posibilidades; primera, cuando $eB1$ es capaz de computar todos estos caminos debido a que todos los nodos involucrados forman parte del mismo dominio y este LSR tiene acceso al grafo completo de la red. Segunda, cuando $eB1$ no es capaz de computar estos caminos porque el dominio está dividido en diferentes áreas IGP. En este caso $eB1$ podrá recibir esta información del modo explicado en la siguiente sección.

Una vez obtenidos estos caminos, $eB1$ los incluye en el mensaje de PATH, en el correspondiente objeto DPMMO, y lo reenvía a uno de los nodos de entrada del siguiente dominio, e.g. $eA1$.

Este LSR, $eA1$, puede obtener, al igual que $eB1$, los distintos caminos disjuntos en el interior de su dominio, $eA1-eD1$ y $eA1-eD3$ por un lado, $eA2-eD1$ y $eA2-eD3$ por otro y, por último, $eA3-eD1$ y $eA3-eD3$, disjuntos por origen.

Estos caminos pueden ser incluidos en un nuevo objeto DPMMO y añadirlo al mensaje de PATH o bien puede, utilizando los caminos del objeto DPMMO existente en el mensaje de PATH que le llegó, utilizar el algoritmo del capítulo 4 y obtener los caminos disjuntos (por origen) que engloben los dos dominios, $eB1-eD1$ y $eB1-eD3$ por un lado y $eB3-eD1$ y $eB3-eD3$ por otro. Estos caminos se incluyen en un nuevo objeto DPMMO que sustituye al antiguo. De este modo se consiguen compactar los datos sin perder información.

[2]En el caso de haber sólo un área interior esta tarea es trivial, si el AS es multiárea se puede utilizar el mecanismo explicado en la sección 5.3.

[3]Nótese que cada grupo de caminos disjuntos desde un LSR de entrada a los diferentes LSRs de salida tiene el mismo color en la figura, y diferente al resto de colores. Estos son los caminos que sirven de entrada al algoritmo del capítulo 4.

Una vez realizado esto el mensaje de PATH llega a $eD1$. Este último LSR calcula, gracias al algoritmo y la información contenida en el mensaje de PATH, los dos caminos disjuntos a utilizar entre $< sO1, sO2 >$ y $< eD1, eD2 >$, añade los caminos computados por el LSR O y añade los que él compute entre $eD1$ y D y $eD2$ y D. Así obtiene los dos caminos disjuntos extremo a extremo. A continuación envía esta información a D en dos objetos RSVP-TE de tipo RRO. D puede responder con dos mensajes de RESV para establecer los dos caminos, el principal y el de respaldo, disjuntos extremo a extremo, o bien, siguiendo un procedimiento más convencional, enviar los dos objetos RRO a O y que sea este quien inicie la señalización (mensaje de PATH + RESV) para establecer cada uno de los caminos.

Resumiendo, el LSR de origen calcula caminos disjuntos a los nodos de entrada del siguiente dominio. A partir de ese momento, un LSR de entrada de cada dominio computa los grupos de caminos disjuntos desde los nodos de entrada del primer dominio a los nodos de entrada del siguiente dominio del LSR que está computando (o se añade la información a tramos). A continuación, el LSR de entrada del último dominio, con la información de que dispone, calcula los dos caminos disjuntos extremo a extremo. Finalmente, el nodo destino establece los caminos computados (o envía los caminos computados al origen para que sea este quien inicie el establecimiento).

Es posible que en alguna ocasión un LSR de entrada no sea capaz de computar tres caminos disjuntos desde un LSR de entrada a los tres LSRs de entrada del siguiente dominio al conjugar dos dominios. Esto puede ocurrir porque el número de LSRs interdominio entre el primer y segundo sea menor, e.g. dos, al número de LSRs de salida, e.g. tres, tal y como se muestra en la figura 5.2. Se puede ver que al tener que calcular *i1* los caminos disjuntos desde *e1* a *s1*, *s2*, y *s3* utilizando la información facilitada por e1, no es posible computar tres caminos disjuntos. En ese caso se deben obtener los tres caminos "lo más disjuntos posibles" o añadir la información, sin poder resumirla.

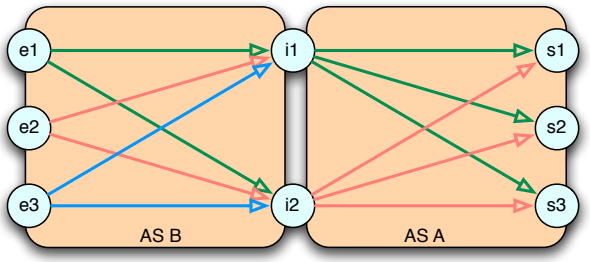


Figura 5.2: Caminos disponibles para obtener caminos unificados a los dos dominios.

Como quedó dicho en el capítulo 4, en el que se explica el algoritmo, es suficiente con que estos sean "lo más disjuntos posibles". $i1$ puede computar los caminos de modo que se mantengan todas las posibilidades marcadas en la figura 5.2, y así

poder resumir los datos a enviar sin perder información. Por ejemplo, si se eligen los caminos tal y como se muestra en la figura 5.3 se conserva toda la información relevante.

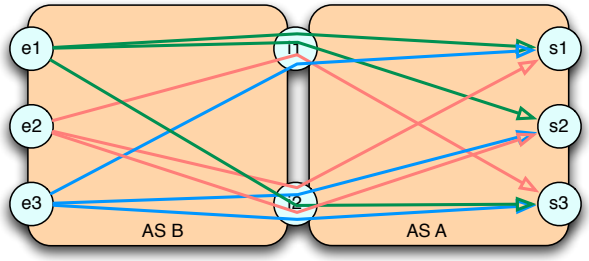


Figura 5.3: Caminos unificados de dos dominios.

De este modo, con los caminos parciales y utilizando el algoritmo es posible obtener un par de caminos disjuntos entre cualquier par de nodos de $< e1, e2, e3 >$, $< s1, s2, s3 >$.

Tal y como ya se ha comentado, este resumen de datos no es más que una optimización, ya que es posible que cada LSR de entrada simplemente incluya los caminos de su dominio como un objeto DPMMO extra, no como sustitución del anterior. De este modo, el LSR destino, *D*, dispondrá de todos los cómputos parciales en lugar de un resumen. Esto, que implica paquetes de señalización mayores, es más eficiente en consumo de tiempo y recursos en los LSRs intermedios, ya que estos se limitan a incluir los caminos, que además pueden tener precomputados.

Enviar la información de los caminos interiores al resto de dominio puede considerarse una política no correcta en muchos dominios. En el siguiente apartado se propone un mecanismo de actuación para evitar tener que compartir este tipo de información entre dominios.

5.2.1. Ocultación de información interior

En el mecanismo explicado anteriormente los dominios comparten entre sí cierta información topológica de sus dominios con el resto de ellos durante su ejecución. Esto, en la mayoría de los casos, no es adecuado y es deseable poder evitarlo.

En realidad, para poder construir los caminos disjuntos finales, no es necesario que un LSR (origen o destino) disponga de toda la información, basta con que cada LSR de entrada al dominio sea capaz de hacerlo. La decisión que se debe tomar globalmente es cuáles son los nodos interdominio que se deben elegir para los caminos principal y de respaldo que permitan establecer los dos LSPs de modo que sean disjuntos y sin tener que recurrir a mecanismos de *cranckback*.

Para ello, al LSR *D* de la figura 5.1 le basta con saber dónde empiezan y terminan los caminos disjuntos computados por los LSR de entrada. Por ejemplo, *D* con la información de que *eB*1 es capaz de computar unos caminos disjuntos hasta *sB*1, *sB*2 y *sB*3, y que *eB*3 es capaz de hacerlo hasta *sB*1 y *sB*2 ya sabe que entre los pares de nodos de entrada *eB*1 y *eB*3 hasta los pares de nodos de salida *sB*1 y *sB*2 se pueden establecer un par de caminos disjuntos (el algoritmo explicado en el capítulo 4 es capaz de asegurar esto). Sin saber cuáles son los caminos concretos, sí puede saber que entre esos dos pares de nodos es posible obtenerlos.

Es por este motivo que los LSRs de entrada de los dominios no tienen por qué dar la información de cuáles son los caminos computados, es suficiente con poner en qué LSRs de salida terminan. Este tipo de información, los LSRs de entrada y salida, es pública, puesto que son nodos interdominio y no hay mayor problema en añadirla en los mensajes interdominio.

Una vez que *D* dispone de toda la información ofrecida por los LSRs de entrada de cada dominio puede seleccionar los LSRs interdominio por los que deben pasar los caminos principal y de respaldo de modo que sean disjuntos y computados sin caer en trampas topológicas.

D envía esta información al origen, *O*, y este inicia la señalización de los caminos principal y de respaldo informando de las parejas de LSRs seleccionados en cada salto interdominio. Debido a que los diferentes LSRs de entrada a un dominio disponen de la misma información, ejecutando los mismos algoritmos de cómputo obtienen, de manera independiente, los mismos caminos disjuntos. Cada uno de los LSR utilizará el que le corresponda para construir el tramo de LSP por el interior del dominio hasta el LSR de salida.

El objeto RSVP-TE IDLO (*Inter-Domain LSRs Object*) que contiene estas parejas de LSRs se especifica en el apéndice B.3. Se utiliza un objeto IDLO por cada uno de los Sistemas Autónomos, indicando los cuatro LSRs involucrados y de qué AS forman parte. Los campos *LSR de entrada 1* y *LSR de salida 1* del objeto se rellenan con los LSR de entrada y salida a utilizar en el AS indicado para el camino principal, los *LSR de entrada 2* y *LSR de salida 2* del objeto se rellenan con los LSR de entrada y salida a utilizar en el AS indicado para el camino de respaldo.

En el siguiente apartado, se explicará que ocurre cuando un Sistema Autónomo está dividido en áreas IGP, que limitan el conocimiento de la red, y cómo distribuir la información necesaria a los diferentes LSRs de entrada.

5.3. Dominio AS dividido en áreas OSPF

El mecanismo explicado en el apartado 5.2 bien podría funcionar computando todas las parejas posibles de caminos disjuntos entre los LSRs de entrada y salida. Pero si los LSRs de entrada no disponen de todo el grafo de la red, el dominio está dividido en áreas IGP, esto no sería posible. A continuación se va a exponer un mecanismo interior para que los LSRs de entrada de los dominio obtengan, aún sin disponer del grafo completo, los caminos disjuntos a los LSRs de salida de los diferentes dominios conectados que requiere el mecanismo interdominio explicado anteriormente.

Además, consiguiendo que un LSR de entrada de un dominio contase con los caminos disjuntos desde él hasta los LSRs de salida a otro dominio y pudiese tener esta información de todos los LSRs de entrada común[4] podría computar un par de caminos disjuntos desde cualquier par de LSRs de entrada (del dominio del que llega la petición) a cualquier par de LSRs de salida (a otro dominio).

La red de la figura 5.4 muestra un Sistema Autónomo en el que existen tres áreas OSPF. En el área 1 se encuentran los dos LSRs de entrada al dominio desde el AS anterior. En el área 2 los dos LSRs de salida al AS siguiente. El área 0, o *backbone*, en OSPF es la que interconecta el resto de áreas.

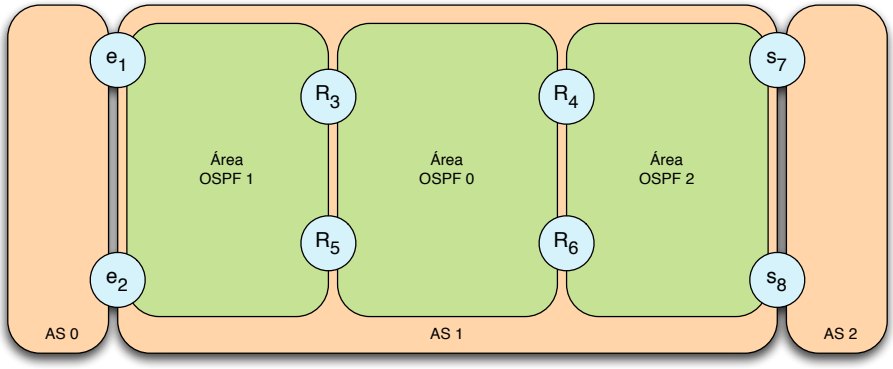


Figura 5.4: División de un Sistema Autónomo en tres áreas.

Para que utilizando OSPF el dominio sea 2-conectado es necesario que haya al menos dos routers de borde entre un área y el área 0, ya que en caso de sólo haber uno si este falla ese área se queda sin conectividad con el área 0 y por consiguiente sin conectividad con el resto del dominio.

En OSPF cada área tiene una base de datos con el grafo completo de su área, aquellos nodos que son frontera de dos áreas tienen el grafo completo de ambas

[4] Aquellos LSRs que son routers de borde con el mismo dominio.

áreas[5]. Todos los routers calculan el SPT (*Shortest-Path Tree*) siendo ellos la raíz del árbol. Los nodos fronteras envían al *backbone* un resumen de alcanzabilidad de su otra área, de modo que los routers en el *backbone* saben cómo llegar a todos los destinos del dominio. En ese momento los routers frontera envían un resumen de alcanzabilidad por medio del área 0 al resto del áreas dentro de su áreas que no sean el *backbone*. De este modo, al final de proceso, cada router de un área tiene el grafo completo de su área e información de alcanzabilidad (y coste) al resto de destinos del dominio por cada uno de los routers frontera de su área.

Con la información disponible en los routers frontera estos pueden computar los caminos disjuntos desde ellos a los routers de borde del dominio (si existe alguno), ya que disponen del grafo de todo el área. Por ejemplo, en la figura 5.5, R_4 puede computar $P_{4,7}$[6] y $P_{4,8}$ disjuntos[7]. Del mismo modo puede hacerlo R_6. Esta información puede repartirse entre todos los routers frontera del *backbone*.

Cada router frontera del *backbone* puede entonces computar los caminos disjuntos desde él a los routers de borde del mismo dominio vecino. Por ejemplo, en la figura, R_3 puede calcular gracias a $P_{4,7}$, $P_{4,8}$, $P_{6,7}$ y $P_{6,8}$ un par de caminos disjuntos dentro del área 1 que le permiten computar (dispone además del grafo de red del área 0) dos caminos disjuntos desde él a los routers de salida del dominio, $P_{3,7}$ y $P_{5,8}$. R_5 actuaría del mismo modo.

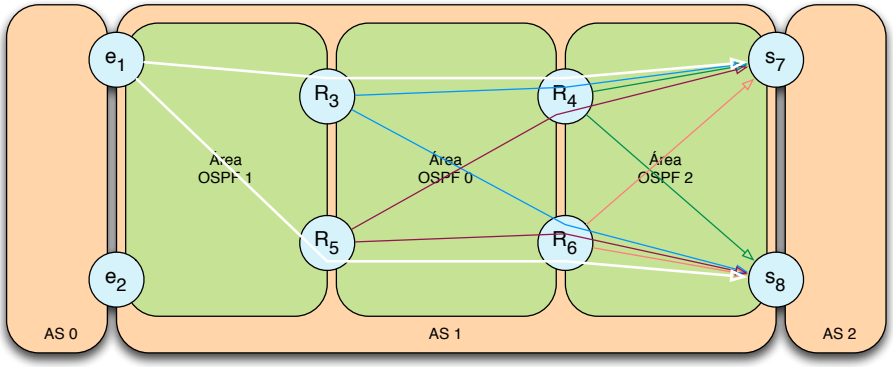


Figura 5.5: Computo y distribución de rutas disjuntas en un AS con tres áreas.

Una vez que los routers frontera del *backbone* tienen computados estos caminos se los envían a los routers de borde a otros dominios que haya en cada una de las

[5]Durante todo el capítulo se distinguirá entre routers frontera y routers de borde. Un router frontera se refiere a un router que pertenece a dos áreas IGP. Un router de borde se refiere a un router limítrofe con otro dominio. Puede suceder que un router frontera sea simultáneamente un router de borde.

[6]$P_{4,7}$ se refiere al camino que va desde el router 4 (R_4) hasta el router 7 (s_7).

[7]Nótese que en las figuras los caminos que son disjuntos entre sí se han marcado del mismo color.

áreas. Al igual que los routers frontera estos routers de borde pueden calcular caminos disjuntos desde ellos a los routers de borde de otro dominio que estén en otras áreas. Por ejemplo, e_1 puede computar gracias a $P_{3,7}$, $P_{3,8}$, $P_{5,7}$ y $P_{5,8}$ los caminos disjuntos $P_{1,7}$ y $P_{1,8}$.

e_1 no sólo puede obtener este par de caminos que atraviesan el dominio entero hasta los routers de borde del siguiente dominio, si no que también puede computar los de e_2 (están en el mismo área). Justamente esta es la información que e_1 necesita para incluir en el menaje de PATH del mecanismo explicado en la sección anterior. Si bien, a continuación, veremos que es mejor opción que cada LSR de borde compute sus propios caminos y los distribuya al resto de LSRs.

Para ello hay que pensar en un escenario más complejo, en el que los routers de borde con otro dominio no están en el mismo área IGP, por ejemplo en la figura 5.6.

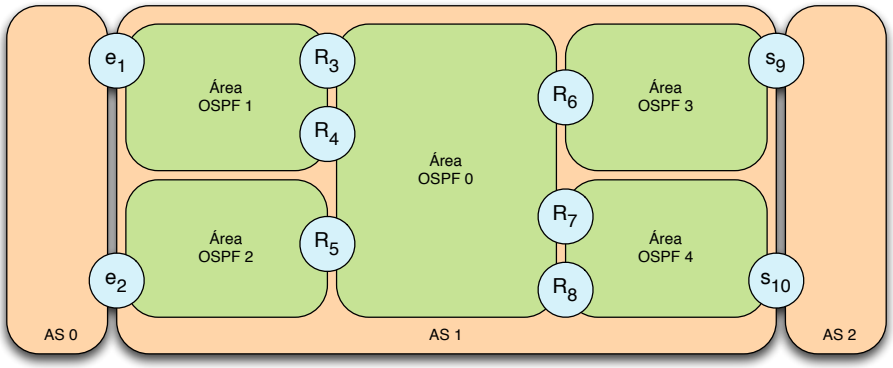


Figura 5.6: División de un Sistema Autónomo en cinco áreas.

De igual modo que antes, e_1, después de todo el proceso, es capaz de computar dos caminos disjuntos hasta los nodos de borde con *AS 2*, $P_{1,9}$ y $P_{1,10}$ (ver figura 5.7). El problema, en este caso, es que al estar en áreas diferentes el propio e_1 no puede computar los caminos desde e_2.

Por este motivo es necesario añadir un paso más al mecanismo explicado arriba. Una vez computados los caminos disjuntos al conjunto de routers de borde de otro dominio, los routers de borde deben enviar estos caminos computados al resto de routers de borde que lo sean del mismo dominio. Este intercambio de información se puede realizar por medio de IBGP (*Interior BGP*). De este modo, al final del proceso, cada uno de los routers de borde con un dominio, e.g. e_1 y e_2 que son routers de borde con *AS 0*, tienen computados los caminos disjuntos a los routers de borde de otro dominio, e.g. s_9 y s_{10} que los son de *AS 2*.

Puede verse que en el caso de los caminos computados por e_2 no serán disjuntos del todo, ya que el tramo dentro del área 2 deberá ser compartido, pero una vez

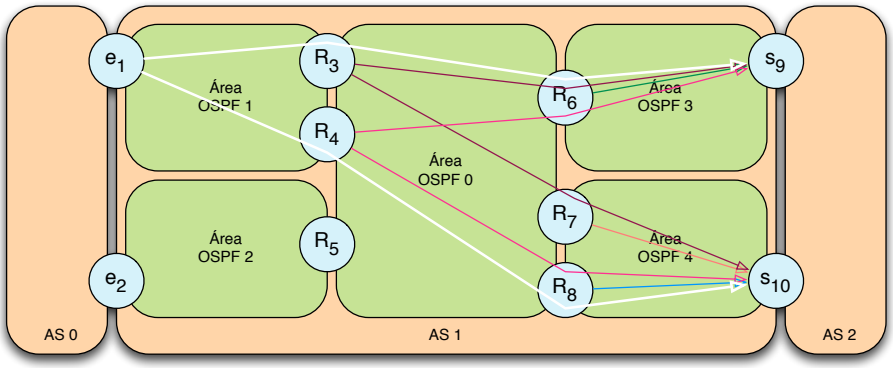


Figura 5.7: Computo y distribución de rutas disjuntas en un AS con cinco áreas.

en el *backbone* ambos caminos son disjuntos hasta el final. Ya se vio, cuando se explicaron los ejemplos del algoritmo, que realmente bastaba con que los caminos fuesen "lo más disjuntos posible". Cualquier cruce entre los caminos de e_1 y e_2 se producirá fuera del área 2 lo que no interfiere negativamente en el cómputo de la pareja de caminos disjuntos.

Otra característica de este modo de proceder es que los caminos los computa cada uno de los routers de borde y este los distribuye al resto de routers. La ventaja consiste en que este mecanismo, al estar muy ligado al algoritmo presentado en el capítulo 4, respeta las políticas locales de cómputo de cada router de entrada. Esta característica también fue resaltada como positiva en el análisis que se hizo del algoritmo (sección 4.4.3).

Resumen del mecanismo de obtención de caminos disjuntos de un nodo a los nodos de salida en un dominio multiárea. El mecanismo se puede dividir en 6 pasos:

1. Los routers frontera entre el área k y el *backbone* computan el *grupo* de caminos disjuntos desde ese router al conjunto de routers de borde que pertenezcan al área k y que comparten el mismo dominio vecino.

2. Esta información es intercambiada entre los routers frontera del *backbone* mediante OSPF.

3. Cada router frontera del backbone computa, gracias a la información recibida, el *grupo* de caminos disjuntos desde ese router al conjunto de routers de borde que comparten el mismo dominio vecino independientemente del área al que pertenezcan.

4. Cada router frontera entre el área k y el *backbone* distribuye esta información a los routers de borde que pertenezcan al área k mediante RSVP-TE.

5. Cada router de borde computa, gracias a la información recibida, el *grupo* de caminos disjuntos desde ese router al conjunto de routers de borde que comparten el mismo dominio vecino independientemente del área al que pertenezcan.

6. Finalmente, cada router de borde envía los caminos computados al resto de routers de borde con los que comparte dominio vecino mediante OSPF.

Estos mensajes intercambiados entre los routers pueden enviarse mediante un LSA (*Link State Advertisement*) de OSPF. El problema reside en que no existen mensajes LSA que permitan enviar secuencias de nodos y enlaces para informar de un camino. Sería necesario definir un nuevo tipo de LSA, el P-LSA (*Path-LSA*). Este sería similar al R-LSA (*Router-LSA*), que permite informar sobre un conjunto de enlaces en el mismo LSA [4], pero en el que además de enlaces se pueden incluir nodos.

Podría también pensarse en utilizar mensajes RSVP-TE para trasnportar esta información por ser información sobre caminos, pero en realidad la información intercambiada es utilizada para conocer si es posible realizar encaminamiento disjunto simultáneo, es decir, información de encaminamiento, por lo que nos parece más adecuado utilizar el protocolo de encaminamiento, en este caso OSPF, para este menester que un protocolo de reserva de recursos como es RSVP.

5.4. Caminos disjuntos por parejas en lugar de por *grupos*

Puede ocurrir que en alguna ocasión los nodos de salida contra los que hay que calcular los caminos disjuntos desde el mismo nodo de entrada sean más de dos, como en la figura 5.8.(a). En ese caso, aunque existan las diferentes soluciones esto no asegura que los tres (o más) caminos a computar sean disjuntos entre sí. Puede ser necesario computar los caminos disjuntos por parejas de nodos. En la figura 5.8.(b) se puede ver el caso en el que hay que calcular dos parejas diferentes. Podrían ser necesarias tres, pero si $P^a_{1,1}$ y $P^b_{1,3}$ son disjuntas estas podrían hacer de $P^c_{1,1}$ y $P^c_{1,3}$ sin ser necesario ponerlas explícitamente.

Para incluir en los mensajes de PATH, e_1, en lugar de un objeto DPMMO puede necesitar utilizar más de uno. Al tener todos ellos el mismo nodo de origen queda claro que son diferentes *grupos* de caminos que forman parte del mismo conjunto.

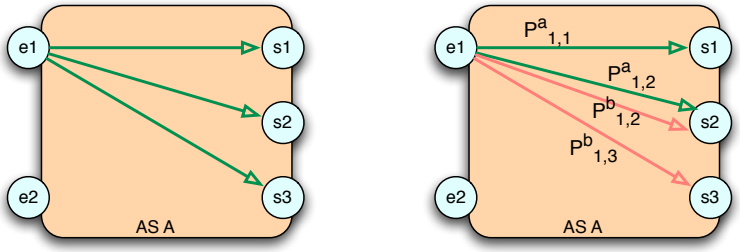


Figura 5.8: Caminos disjuntos de un nodo de entrada a varios nodos de salida en grupo o por parejas.

5.5. Conclusiones

En este capítulo se ha presentado un mecanismo para el cómputo de caminos disjuntos extremo a extremo que crucen por por la misma secuencia de Sistemas Autónomos. Para ello, el mecanismo está basado en el algoritmo presentado en el capítulo 4. También se ha explicado el mecanismo intradominio que hay que llevar a cabo en un dominio dividido en diversas áreas IGP para obtener y distribuir la información de routing necesaria. Una vez que este mecanismo interior está en marcha, los LSRs de entrada a los diferentes dominios disponen de la información requerida en el procedimiento de cómputo del mecanismo interdominio.

Este mecanismo interdominio puede establecer los caminos principal y de respaldo con un solo intercambio de mensajes de señalización (PATH + RESV). En este caso, se ha visto, que los caminos computados en cada uno de los distintos dominios deben ser enviados en los mensajes de PATH y que, por tanto, es necesaria la compartición de información topológica entre los diferentes dominios.

A continuación se ha explicado que con el coste de utilizar dos intercambios de mensajes de señalización, el primero para obtener los nodos interdominio a utilizar y el segundo para establecer los caminos principal y de respaldo, es posible computar y establecer los caminos disjuntos interdominio extremo a extremo sin necesidad de compartir información topológica entre dominios.

En cualquiera de ambos casos, por el hecho de utilizar el algoritmo explicado en el capítulo 4, es posible afirmar que si existe solución al problema este mecanismo es capaz de encontrarla en primera iteración, sin caer en trampas topológicas.

Se puede decir, entonces, que, hasta donde llega nuestro conocimiento, este mecanismo es el primero propuesto para el cómputo y establecimiento de LSPs interdominio disjuntos que es distribuido, evita las topologías trampa, permite la ocultación de las topologías entre dominios, tiene en cuenta la división en áreas IGP de los dominios y es escalable que se ha propuesto.

Capítulo 6

Protegiendo únicamente la zona interdominio

En este capítulo se va a estudiar un mecanismo de respaldo interdominio que utiliza varios LSPs de protección complementarios para proteger el LSP principal. Un LSP de respaldo interdominio extremo a extremo para proteger los nodos y enlaces interdominio y LSPs locales, internos a cada dominio, para proteger nodos y enlaces en el interior de un dominio concreto.

Para ello primero se explicarán los diferentes tipos de esquemas de respaldo intradominio existentes y cómo todos ellos son capaces de proteger todos los recursos interiores de un dominio. Una vez vistos los recursos que el respaldo intradominio no es capaz de proteger se propone un LSP de respaldo interdominio que los proteja.

Seguidamente se analiza el consumo de recursos utilizando este esquema, teniendo en cuenta que dentro de un dominio hay dos tipos de LSPs de respaldo, los "*locales*" al dominio y el interdominio.

Finalmente, se explica la señalización necesaria para establecer los LSPs principal y de respaldo utilizando un nuevo objeto RSVP-TE, el IDRO. Para concluir, un resumen de lo expuesto y unas conclusiones.

6.1. Introducción

Ya se ha visto que hay dos vías fundamentales para proteger un LSP interdominio. Establecer el LSP de respaldo que pase por la misma secuencia de Sistemas Autónomos que el LSP principal o establecerlo por una secuencia de Sistemas Autónomos diferentes. En el primer caso hay que procurar que los caminos elegidos para ambos LSPs sean disjuntos. En el segundo caso estos caminos son intrínsecamente

disjuntos.

En este capítulo estudiaremos un mecanismo de respaldo válido para ambos escenarios, pero que adquiere plena funcionalidad en el primero de ellos. La principal característica de este esquema es que los dos LSPs interdominio pueden compartir parte del camino, relajando las restricciones de caminos disjuntos y así facilitando su cálculo. Tiene además la ventaja de que si en los diferentes dominios existen mecanismos de protección intradominio en uso es posible optimizar el consumo de recursos en los dominios respecto a otros esquemas. En cambio, si estos mecanismos no existen el consumo de recursos resulta algo superior que en otros esquemas.

6.2. Divide y vencerás

En esta sección se van a estudiar los dos tipos LSPs de respaldo utilizados en el esquema de respaldo propuesto en este capítulo; LSP de respaldo intradominio y LSP de respaldo interdominio.

6.2.1. Recuperación intradominio de un LSP interdominio

La forma más rápida de proteger un LSP ante el fallo de un enlace o un nodo es utilizando un mecanismo de recuperación local, después recuperación de segmento y, finalmente, recuperación global o extremo a extremo.

La ventaja de la recuperación global es que un único LSP de respaldo protege ante el fallo de cualquier elemento del LSP principal. En la recuperación local es necesario un LSP de respaldo por cada enlace y por cada nodo que se quiera proteger. Un compromiso entre ambos esquemas es proteger por segmentos (para más detalle véase la sección 2.2).

Las propuestas de recuperación interdominio actuales son propuestas de tipo global, se intenta proteger todo el LSP principal con un único LSP de respaldo extremo a extremo. Esto obliga a que todo el LSP de respaldo sea disjunto al principal. Realizar este cómputo mediante esquemas distribuidos "tradicionales" tiene los problemas discutidos en la sección 2.4.

Sin embargo, en recuperación interdominio, también es posible utilizar esquemas de recuperación "*local*". "*Local*" aquí significa el interior de un dominio. La recuperación local intradominio se refiere a un un enlace o a un nodo. En este caso con recuperación "*local*" para un LSP interdominio se refiere a la recuperación que de manera local al dominio se realiza sobre el segmento del LSP interdominio que corresponde con dicho dominio.

Se va a ver con más detalle qué ocurre cuando se realiza recuperación global,

de segmento y local dentro de un dominio para proteger el segmento de un LSP principal interdominio que transita por dicho dominio.

Recuperación global En la figura 6.1 se puede ver un ejemplo de recuperación del segmento del LSP interdominio *LSP 1* en el dominio *Dominio 2*. El LSP de respaldo, *B1*, es un LSP global, extremo a extremo dentro del dominio que protege todo el segmento interno del LSP principal.

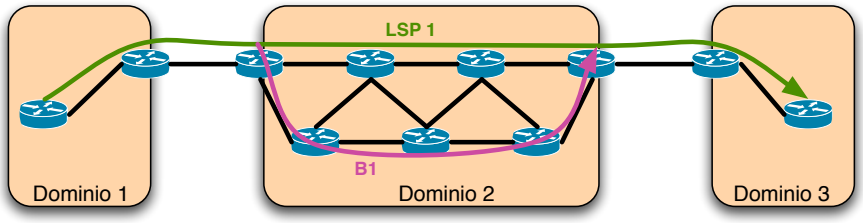


Figura 6.1: Ejemplo de LSP de recuperación global intradominio.

Recuperación de segmento En la figura 6.2 en este caso muestran dos LSPs de recuperación, *B1* y *B2*. Cada uno de ellos respalda un sub-segmento del LSP interdominio, *LSP 1*, de modo que entre ambos son capaces de recuperar cualquier fallo de *LSP 1* dentro del dominio.

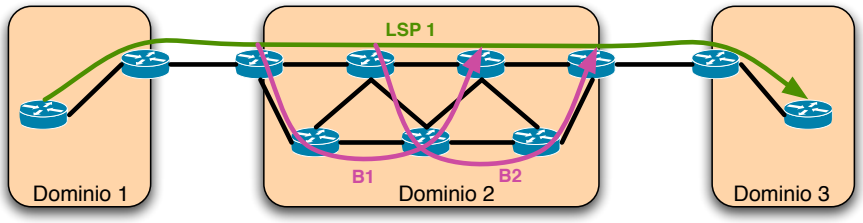


Figura 6.2: Ejemplo de LSPs de recuperación por segmentos intradominio.

Recuperación local Por último, en la figura 6.3, se muestra el caso en el que se realiza recuperación local dentro del *Dominio 2*. En este caso son necesarios cinco LSPs de recuperación, *B1* - *B5*, tal y como se vio en la sección 2.3 que es necesario para realizar *fast reroute*. Cada uno de ellos recupera el fallo de un nodo, *B1* y *B2*, o un enlace, *B3*, *B4* y *B5*.

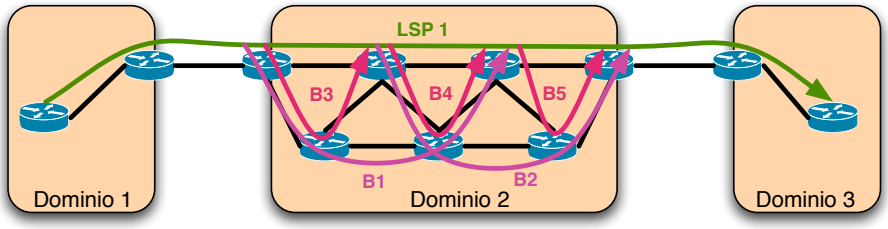


Figura 6.3: Ejemplo de LSPs de recuperación local intradominio.

Como se puede ver hay estrategias que utilizan más LSPs de respaldo que otras. El compromiso está entre mayor o menor velocidad de reacción y mayor o menor número de LSPs necesarios. Cuantos más LSPs de respaldo se utilicen existirá un PSL y un PML más próximos al fallo que permitirán una recuperación más rápida.

Para cada una de estas estrategias de recuperación se puede utilizar tanto reencaminamiento como protección (las ventajas y desventajas de uno y otro se analizaron en el apartado 2.2). Incluso se pueden utilizar otras estrategias, como, por ejemplo, Makam [56] o Hasking [57], en lugar de estas más sencillas.

Tal y como se puede ver en todos los casos, los LSP de respaldo son capaces de proteger todos los nodos y enlaces del LSP interdominio principal. Lo único que no se capaz de proteger "*localmente*" son los propios nodos interdominio. Estos nodos y enlaces interdominio hay que protegerlos mediante recuperación interdominio.

En las disertaciones que siguen se utilizará el esquema de recuperación mostrado en la figura 6.1, *recuperación global intradominio*, como ejemplo cuando haga falta referirse a un esquema de protección "*local*", aunque cualquiera de los posibles esquemas de recuperación "*local*" en el intradominio para el LSP interdominio pueda ser utilizado. Si en algún ejemplo se requiere concretar sobre un caso específico se citará el caso explícitamente. De este modo, se pretende aislar la problemática interdominio de la recuperación intradominio concreta utilizada. Se considerará que el esquema específico de recuperación intradominio es capaz de proteger el LSP interdominio principal de cualquier fallo dentro del dominio, excepto de un fallo en los nodos frontera y en los enlaces interdominio, tal y como se ha visto en los tres ejemplos mencionados arriba.

6.2.2. Recuperación interdominio de un LSP interdominio cuando existe recuperación local intradominio

A pesar de utilizar recuperación "*local*" dentro de los dominios ya se ha visto que los nodos y enlaces interdominio quedan sin proteger. Es necesario entonces un LSP de respaldo interdominio extremo a extremo.

Por el hecho de seguir necesitando realizar un un LSP de respaldo interdominio pudiera parecer que no se ha mejorado nada, incluso, empeorado la situación por tener que realizar recuperación "*local*". En realidad, lo que se ha mejorado es que el LSP de respaldo interdominio sólo necesita proteger la zona interdominio, no todo el LSP principal. A esto se une el hecho de que los dominios en realidad sí protegen sus enlaces y nodos y con este esquema se está aprovechando esta circunstancia.

En la figura 6.4 se muestra un ejemplo. Como se puede ver el LSP de respaldo, *BG1* comparte parte del camino con el LSP de respaldo "*local*", *B1*. Pero no es un problema, puesto que no es necesario que sean disjuntos. Cuando falla un recurso interno del dominio entra en funcionamiento la recuperación "*local*" del dominio, y se utiliza *B1*, en el caso de un fallo interdominio se utilizaría *BG1*. Por lo que *BG1* y *B1* pueden compartir camino físico y recursos (en el caso de que los LSPs de respaldo estén preestablecidos) ya que no se utilizarán simultáneamente.

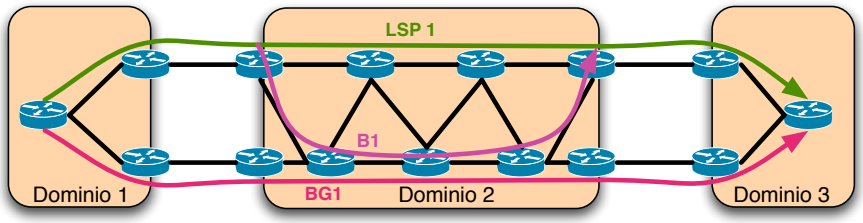


Figura 6.4: Ejemplo de LSP de respaldo interdominio cuando existe recuperación intradominio.

Debido a esta particularidad tampoco es necesario que el LSP principal y el de respaldo interdominio sean completamente disjuntos. Sólo en los nodos y enlaces interdominio, puesto que durante los fallos en recursos internos de un dominio la recuperación se realiza con los LSPs de respaldo "*locales*". Si el fallo es interdominio entonces se utilizará el LSP de respaldo interdominio, *BG1*, pero puesto que no ha fallado nada en el dominio los recursos internos de estos pueden ser compartidos.

En la figura 6.5 se presenta un ejemplo en el que el LSP de respaldo interdominio comparte parte del camino de respaldo "*local*" en el *Dominio 2*. Si hubiese un fallo en el enlace $R_5^2 - R_6^2$ bastaría con notificar a R_1^2 para utilizar el LSP de respaldo *B1*, quedando solucionado un problema a nivel interno por el propio dominio. En cambio, un fallo en el nodo R_1^2 provocaría una notificación de R_3^1 hacia O para conmutar al LSP de respaldo extremo a extremo, camino disjunto al LSP principal en el recurso fallido.

De este modo se simplifica mucho la búsqueda de un LSP de respaldo extremo a extremo, puesto que este y el LSP principal no deben ser disjuntos en todo el camino.

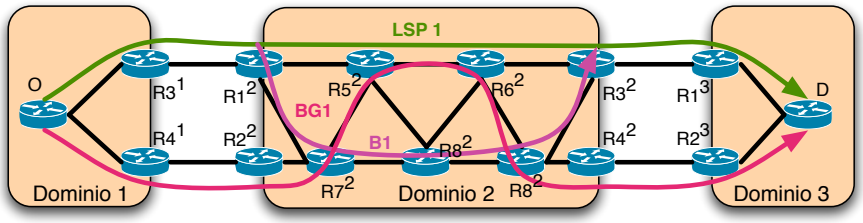


Figura 6.5: Ejemplo de LSP de respaldo interdominio cuando existe recuperación intradominio. El LSP principal y de respaldo comparten parte del camino.

De hecho no es necesario que sea disjunto en el interior de los diferentes dominios, con lo que la posibilidad de caer en una trampa topológica[1] interna es nula si los nodos de entrada y salida de los dominios en el LSP principal y de respaldo son distintos[2].

Cuanto más tramo de camino compartan el LSP de respaldo interdominio con el LSP principal y los de respaldo "*locales*" mayor eficiencia habrá en el uso de recursos debido a que podrán compartir reservas. Esto es posible, por un lado, porque el LSP principal y el de respaldo interdominio nunca se utilizarán simultáneamente y debido a que el camino "*local*" está protegido por los LSPs "*locales*" serían estos últimos los utilizados; y por otro, porque el LSP de respaldo interdominio y los LSPs de respaldo "*locales*" protegen zonas distintas, luego el primero de ellos puede reutilizar los recursos de los últimos.

A continuación se analizarán algunos de los aspectos más importantes de esta propuesta.

6.3. Análisis

En el esquema de recuperación propuesto en este capítulo existen dos LSPs de respaldo para cada LSP principal dentro de cada dominio. Esto, en una primera impresión, puede parecer menos eficiente en el uso de recursos que cualquiera de los esquemas revisados en la sección 2.4 o de los propuestos en esta Tesis Doctoral, en los que sólo existe un LSP de respaldo, pero no tiene porqué ser así. Si se tiene en cuenta que los dominios que implementan una red MPLS tienen algún mecanismo de recuperación de sus propios enlaces es posible utilizar esta protección intradominio para los recursos internos del dominio optimizando el uso de los recursos internos.

[1] Véase el apartado 2.4.4.

[2] Se presupone que la red es totalmente accesible, esto es, cualquier nodo de entrada es capaz de conectarse con cualquier nodo de salida.

Esta protección interna se llevará a cabo en cualquier caso (se tenga en cuenta o no en la recuperación interdominio), debido, además, a que los dominios prefieren no dar a conocer a terceros fallos internos de su red y reparlos internamente. Por tanto, a la hora de diseñar un mecanismo de recuperación interdominio es más eficiente tener esto en cuenta.

El LSP de respaldo interdominio se ha visto que es necesario para proteger los recursos interdominio. Tener en cuenta la protección "*local*" existente en los dominios a la hora de planificar el LSP de respaldo interdominio posibilita no sólo no consumir más recursos, sino ahorrar en ellos. Cuantos más recursos comparta el LSP de respaldo interdominio con los LSP de respaldo "*locales*" o el LSP principal dentro del dominio menor será el uso de recursos.

Es posible realizar un análisis del coste en el que se incurre por utilizar esta estrategia, con dos LSPs de respaldo, frente a otras estrategias diferentes. En la figura 6.6 se muestra un dominio en el que se destacan únicamente los LSRs relevantes. Los dos LSRs de entrada utilizados, uno para el LSP principal, E_1, y otro para el de respaldo, E_2. También los LSRs de salida del dominio utilizados en el LSP principal y el de resplado, S_1 y S_2 respectivamente. Se va a estudiar los casos mejor y peor, por lo que del interior sólo son necesarios tres caminos alternativos. Supuestos costes unitarios, los enlaces n_1, n_2 y n_3 representan un camino de LSRs y enlaces con coste total n_1, n_2 y n_3 respectivamente, y además, $n_1 < n_2 < n_3$. Así pues, el coste el LSP que pasase por E_1 - I_1 - I_4 - S_1 sería de $1+n_1+1 = 2+n_1$.

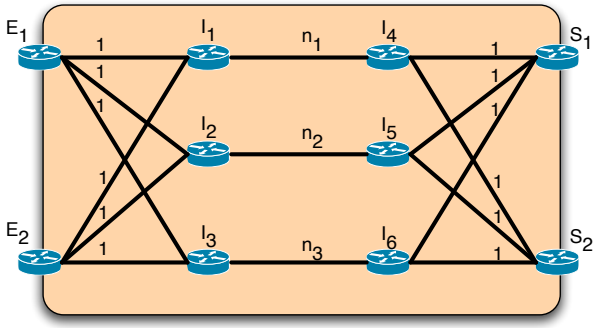


Figura 6.6: Coste en recursos utilizados de los LSPs dentro de un dominio

A continuación se van a estudiar las diferentes situaciones posibles dentro de un dominio. Se supone que el LSP principal siempre se establece siguiendo el camino E_1 - I_1 - I_4 - S_1, y que por tanto su coste es $2+n_1$.

En la figura se han especificado los primeros y últimos enlaces dentro del dominio para mostrar las diferentes posibilidades. Debido a esto el coste resultante de un camino que atraviese el dominio es $2+n_i$. Para simplificar se utilizara la siguiente notación: $n'_i = 2+n_i$.

En este estudio se va a suponer que el LSP principal siempre se establece por el camino óptimo, si bien en algunos esquemas esto no es posible al requerir que el LSP principal y de respaldo sean completamente disjuntos. Como ya se ha dicho este inconveniente no es tal en el esquema que aquí se propone, puesto que el LSP principal y el de respaldo no necesitan ser disjuntos en el interior del dominio y se establecen de manera independiente.

Caso 1. LSP principal y de respaldo disjuntos sin protección "*interior*". Este caso se corresponde con el coste que se tendría en la mayoría de los esquemas interdominio.

El LSP de respaldo se establecerá por E_2 - I_2 - I_5 - S_2, con un coste total de $2+n_2$.

Por tanto el coste total de este caso (sumando el coste de todos los LSPs):

$$C_T^1 = 4 + n_1 + n_2 \tag{6.1}$$

Caso 2. LSP principal y de respaldo disjuntos con protección "*interior*" disjunta. Este sería el caso peor, se utiliza protección interior pero no se tiene en cuenta y el LSP de respaldo interdominio se computa disjunto a ambos.

El LSP de respaldo interior se establecerá por E_1 - I_2 - I_5 - S_1, con un coste total de $2+n_2$.

El LSP de respaldo interdominio se establecerá por E_2 - I_3 - I_6 - S_2, con un coste total de $2+n_3$.

Por tanto el coste total de este caso (sumando el coste de todos los LSPs):

$$C_T^2 = 6 + n_1 + n_2 + n_3 \tag{6.2}$$

Como se puede ver se corresponde con el caso peor.

Caso 3. Existe protección interior y se tiene en cuenta, el LSP de respaldo interdominio se computa lo más solapado posible al LSP principal o al LSP de respaldo "*interior*". Este caso se corresponde con el caso óptimo de utilización del esquema de recuperación propuesto en este capítulo.

El LSP de respaldo interior se establecerá por E_1 - I_2 - I_5 - S_1, con un coste total de $2+n_2$.

El LSP de respaldo interdominio se establecerá por E_2 - I_1 - I_4 - S_2, con un coste total de $2+n_1$., pero si se comparten los recursos entre ambos LSPs el coste real será de 2.

Por tanto, el coste total de este caso (sumando el coste real consumido de todos los LSPs):

$$C_T^3 = 6 + n_1 + n_2 \tag{6.3}$$

Se puede suponer que $n_1 \approx n_2 \approx n_3 = n$, lo que indica que el número medio de saltos para atravesar el dominio son $n+2$. Utilizar esta simplificación no es descabellada facilita los cálculos.

Es posible, entonces, reescribir las ecuaciones 6.1, 6.2 y 6.3 como:

$$C_T^1 = 2n' \tag{6.4}$$

$$C_T^2 = 3n' \tag{6.5}$$

$$C_T^3 = 2 + 2n' \tag{6.6}$$

En el caso mejor, el aumento de coste es sólo de $\Delta C_T = C_T^3 - C_T^1 = 2$, independientemente del tamaño de la red. En el caso peor es de $\Delta C_T = C_T^2 - C_T^1 = n'$, o lo que es lo mismo, aproximadamente igual al número de saltos necesarios para atravesar la red. Por tanto, el aumento en uso de recursos respecto al caso base varía entre

$$\frac{\Delta C_T}{C_T^1} = \frac{2}{2n'} = \frac{\mathbf{1}}{\mathbf{n'}} \tag{6.7}$$

y

$$\frac{\Delta C_T}{C_T^1} = \frac{n'}{2n'} = \frac{\mathbf{1}}{\mathbf{2}} \tag{6.8}$$

También puede calcularse la relación entre los recursos utilizados por el LSP de respaldo interdominio en el interior del dominio y los recursos utilizados por el LSP principal en el interior del dominio.

En el caso de tener dos LSP disjuntos extremo a extremo, el principal y el de respaldo, el aumento de recursos es del 100 %. Este es el caso de los esquemas de respaldo interdominio vistos en la literatura:

$$\frac{C_{LSPrespaldo}}{C_{LSPprincipal}} = \frac{n'}{n'} = 1 \tag{6.9}$$

En cambio, si sí se tiene en cuenta el mecanismo de respaldo intradominio, y se considera que los recursos consumidos por este son fijos, tenemos dos casos. Cuando no se comparte ningún recurso interior (caso peor):

$$\frac{C_{LSPrespaldo}}{C_{LSPprincipal}} = \frac{n'}{n'} = 1 \tag{6.10}$$

Cuando sí se comparten recursos entre el LSP principal, el de respaldo "*local*" y el de respaldo interdominio. Entonces, el aumento en uso de recursos puede verse disminuido hasta (caso mejor):

$$\frac{C_{LSPrespaldo}}{C_{LSPprincipal}} = \frac{2}{n'} \tag{6.11}$$

(Nótese que si, por ejemplo, el LSP de respaldo interdominio comparte todo el camino interior del dominio con el LSP de respaldo "*interior*" el aumento en el uso de recursos dentro de la red por culpa del LSP de respaldo interdominio es solamente de 2)

Con lo que la mejora puede llegar a ser de:

$$1 - \frac{2}{n'} = \frac{\textbf{n'-2}}{\textbf{n'}} \tag{6.12}$$

Si, por ejemplo, el número medio de saltos para atravesar una red es 5, la mejora aproximada puede llegar a ser de:

$$\frac{n'-2}{n'} = \frac{3}{5} = \textbf{60 \%} \tag{6.13}$$

Es decir, que se pueden llegar a ahorrar un 60 % de los recursos para construir el LSP de respaldo interdominio dentro del dominio en cuestión utilizando el mecanismo propuesto en este capítulo, ya que tiene en cuenta la existencia de los LPS de protección intradominio.

Cabe destacar que si bien los cálculos realizados son aproximados se ha supuesto el esquema de respaldo intradominio que menos recursos consume. Es probable que con otros esquemas (de segmento o globales) dentro del dominio la mejora sea mayor.

Tenemos, por un lado, que el aumento de consumo de recursos puede llegar a ser del 50 %, pero si se tiene en cuenta que los recursos para el respaldo "*local*" ya están siendo utilizados, se construyan los LSP interdominio o no, entonces podemos considerar que teniendo en cuenta y utilizando estos LSPs interiores se pueden ahorrar recursos (hasta un 60 % en una red de 5 saltos).

Hay que hacer notar que el cálculo del LSP de respaldo interdominio se simplifica, puesto que el único requisito es que no compartan los recursos interdominio, sin restricciones en el interior de cada dominio. Además, la información de los recursos interdominio es conocida por el resto de dominios, es pública, por lo que no es necesario compartir información privada entre los dominios, lo único importante es el nodo de entrada y salida de cada dominio.

Cabe destacar también que debido a que los LSP principal y de respaldo pueden (y deben) compartir recursos en el interior del dominio no hay restricciones especiales que cumplir. Entonces, si la red es completamente accesible[3], y debido a que el LSP principal y el de respaldo no deben cumplir requisitos adicionales, estos siempre se pueden establecer de manera independiente. No hay posibilidad de caer en trampas topológicas, ya que se establecen dos LSPs de manera independiente, y, por tanto, no es necesario ningún mecanismo de *crankback* para asegurar el éxito en el establecimiento de los LSPs.

En el siguiente apartado se verá cómo utilizar RSVP-TE para señalizar tanto el LSP principal como el de respaldo interdominio.

6.4. Señalización

En este apartado se va a explicar el mecanismo completo de cómputo y señalización que establecen los LSPs principal y de respaldo para una conexión interdominio entre Sistemas Autónomos. Se hará especial hincapié en la señalización utilizada.

Para la señalización se supondrá que los dominios son Sistemas Autónomos, en los que la compartición de información entre ellos es muy limitada y este esquema de recuperación puede ser muy beneficioso. Esto es debido a que no es necesario compartir información extra entre los diferentes dominios y aún así esto no provoca problemas de topologías trampa.

Para el mecanismo de señalización que se va a explicar a continuación es necesario utilizar un nuevo objeto RSVP-TE, el objeto IDRO (*Inter-Domain Routing Object*), detallado en el anexo B.1. Este nuevo objeto especifica cuatro subobjetos de borde de un dominio en el siguiente orden; *recursos de entrada principal*, *LSR de entrada de respaldo*, *recursos de salida principal*, *LSR de salida de respaldo*. Además se incluye el identificador del dominio al que pertenecen los LSRs. Con esta información de cada dominio es suficiente para poder proteger un camino principal interdominio.

En la figura 6.7 se muestra un esquema de los mensajes de señalización intercam-

[3]Se puede alcanzar cualquier nodo desde cualquier otro de la red. Esto lo cumple cualquier red de comunicaciones.

biados durante el cálculo y establecimiento de los caminos principal y de respaldo. Señalización que se va a detallar a continuación.

El LSR *O*, del dominio *AS O*, quiere establecer un LSP con el LSR *D*, del dominio *AS D*. Para ello, primero selecciona el LSR de salida de su dominio que le lleve a *D*. Si el LSR *O* es un LSR de borde, además, puede calcular la lista de dominios que debe atravesar el LSP, en caso contrario, podría averiguarlo o no. Si no es capaz de hacerlo, en el mensaje de PATH para construir el LSP principal no incluirá un objeto ERO con la lista de ASes. En caso de que sí pueda hacerlo, es posible añadirlo, pero no imprescindible. Este objeto ERO no es imprescindible puesto que el AS_PATH ya está prefijado de antemano, es el que han aprendido los Sistemas Autónomos por medio de BGP. El mensaje de PATH seguirá dicho camino (tanto para el LSP principal como para el LSP de respaldo) aunque no lo indique explícitamente el objeto ERO. Un motivo para incluirlo sería si el AS_PATH seleccionado no es el calculado por BGP y se quiere que los LSPs se establezcan por una secuencia de ASes diferente.

Una vez realizados estos cálculos el LSR *O* envía un mensaje RSVP-TE de PATH para establecer el camino principal y obtener la información necesaria que le permita establecer el camino de respaldo.

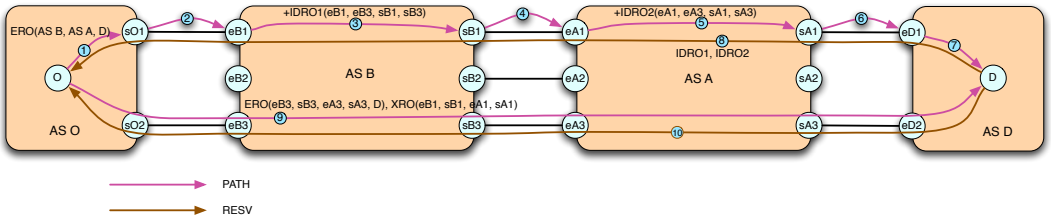


Figura 6.7: Señalización IDRO

Este mensaje de PATH pasa por el LSR de salida del dominio *AS O*, *sO1*. Éste LSR, basado en el destino del mensaje de PATH, determina que el siguiente salto es un LSR de entrada a *AS B*, en el caso del ejemplo de la figura *eB1*. EL LSR de entrada al dominio añade en el mensaje de PATH un nuevo objeto IDRO. En él incluye, además del identificador del dominio, la siguiente información; a sí mismo y el SRLG por el que le llega la petición como *recursos de entrada principal*; debe seleccionar el LSR de salida del dominio que le llevará al destino, puede incluirlo como parte de el campo *recursos de salida principal* o dejarlo para que lo haga este, y encaminar el mensaje de PATH hacia él. Adicionalmente (no es obligatorio), puede seleccionar los LSRs que harán de entrada y salida al dominio para el LSP de respaldo e incluirlos como *LSR de entrada de respaldo* y *LSR de salida de respaldo* en el objeto IDRO. Esta selección la hará el LSR en función de si en el dominio se tiene alguna preferencia por los LSRs a utilizar para el LSPs de respaldo o no. Si

el LSR de entrada principal no hace esta selección esta se irá realizando de manera distribuida durante la señalización del LSP de respaldo.

El mensaje de PATH va de LSR en LSR dentro del dominio *AS B* hasta alcanzar el LSR de salida, *sB1*. Este debe rellenar la información del subobjeto *recursos de salida principal* que estén sin incluir. Una vez realizada esta operación el proceso se repite hasta alcanzar el LSR de entrada y salida del siguiente dominio del siguiente dominio, y así sucesivamente hasta llegar al penúltimo dominio.

Una vez el mensaje de PATH ha llegado a *sA1* este determina utilizar *eD1* como LSR de entrada principal al dominio destino y desde aquí llega al LSR destino.

Cuando el LSR de destino ha recibido el mensaje de PATH este responde con un mensaje RSVP-TE de tipo RESV para establecer el LSP. En dicho mensaje de RESV el LSR *D* copia todos los objetos IDRO, y en el mismo orden, que han llegado en el mensaje de PATH.

Cabe destacar que en el caso de utilizar identificadores globales el SRLG del subobjeto de *recursos de salida principal* de un dominio debería ser el mismo que el SRLG del subobjeto de *recursos de entrada principal*. Si los identificadores son locales al dominio estos se podrán distinguir por el identificador de dominio existente en el objeto IDRO.

Establecido el LSP principal sólo queda calcular y señalizar el LSP de respaldo. Señalizado durante el fallo de algún recurso o previamente a este, esto es, se realice reencaminamiento o protección, el mecanismo es el mismo, sólo cambia el instante de tiempo en que se establece el LSP.

A partir de los objetos IDRO recibidos por el LSR *O* en el mensaje RESV de señalización del LSP principal debe construir dos objetos RSVP-TE nuevos, un de tipo ERO y otro de tipo XRO para el mensaje de PATH de señalización del LSP de respaldo.

El objeto ERO contendrá aquellos LSRs de respaldo que se hayan especificado en los subobjetos *LSR de entrada de respaldo* y *LSR de salida de respaldo* de los objetos IDRO recibidos, ya que son los preferidos por los distintos dominios para establecer el LSP de respaldo. Los que no se especifiquen, será el encaminamiento local a cada nodo el que vaya decidiendo el camino, pero evitando los recursos especificados en el objeto XRO.

El objeto XRO contendrá los LSRs y SRLGs principales especificados en los subobjetos *recursos de entrada principal* y *recursos de salida principal* de los objetos IDRO recibidos, ya que estos no pueden ser utilizados en el LSP de respaldo debido a que son, precisamente, los recursos a proteger.

Con esta información, objetos XRO y ERO, se envía un mensaje de PATH al destino y se espera el mensaje de RESV que establece el LSP de respaldo que es

capaz de proteger los recursos del interdominio.

6.5. Conclusiones

En este capítulo se ha presentado un nuevo esquema de respaldo interdominio en el que se tiene en cuenta que los dominios realizan recuperación "local", esto es, protegen sus propios recursos de manera interna.

Se han mostrado los posibles esquemas de respaldo que se pueden utilizar en el interdominio, y como independientemente del utilizado, estos son capaces de proteger frente al fallo en cualquier recurso del dominio, excepto en los LSRs de borde. Tampoco es posible proteger desde el interior de un dominio los enlaces interdominio.

Debido a que sólo hay que proteger los recursos interdominio se relajan las restricciones que debe cumplir el LSP de respaldo respecto al LSP principal. Sólo es necesario que sean disjuntos en el interdominio. Además, no es necesario compartir información privada interior entre los dominios. Se ha visto también que el LSP principal siempre es el óptimo, puesto que al no tener restricciones dentro de los dominios no hay problema de caer en una trampa topológica y el LSP de respaldo puede establecerse de manera independiente dentro de los dominios.

Se ha realizado un análisis cualitativo y cuantitativo en el consumo de recursos empleando este esquema, resultando en un ahorro de recursos si los LSPs de respaldo "*interior*" y extremo a extremo comparten recursos.

Finalmente, se ha presentado la señalización a utilizar para llevar a cabo el establecimiento de los LSP principal y de respaldo. Para poder llevar a cabo dicha señalización se ha introducido un nuevo objeto RSVP-TE, el IDRO, cuyo detalle se da en el apéndice B.1.

Si en una red hay implantado MPLS en los routers es más que probable que una de las aplicaciones que se utilice sea la protección de enlaces dentro del dominio, y con esta la de LSPs. Suponiendo pues que es el propio dominio quien protege los LSPs (intradominio o interdominio) ante fallos dentro de la red, los LSPs de respaldo interdominio no deberían preocuparse de proteger estos recursos. Sólo deberían preocuparse de aquellos que los dominios por sí mismos no pueden proteger, y requieren de colaboración con otros dominios para ello. Estos recursos son los del interdominio. Por lo que el esquema de respaldo presentado en este capítulo puede ser válido en un gran número de escenarios.

Ante el fallo de un recurso en el interior de un dominio se utilizan los mecanismos de recuperación intradominio, en el caso de un fallo en un recurso interdominio se utiliza el LSP de respaldo interdominio. La ventaja de este esquema es que un

fallo dentro de un dominio no trasciende al resto de dominios.

En los esquemas de respaldo interdominio "tradicionales" un fallo en cualquier nodo o enlace del LSP principal provoca un cambio al LSP de respaldo. Esto requiere que cualquier fallo interno de un dominio se notifique al LSR origen y, por tanto, que la notificación de fallo circule por otros dominios. Típicamente, esto no es deseable por parte de los Sistemas Autónomos, ya que si son reacios a compartir información topológica mucho más a informar sobre fallos en su red.

La simplificación en el cómputo de los LSPs es mucho mayor que en otros mecanismos, debido a que el LSP principal se computa como el LSP óptimo extremo a extremo sin tener en cuenta otros factores. El cómputo del LSP de respaldo también se computa como el LSP óptimo extremo a extremo pero con la única restricción de no compartir los mismos recursos interdominio. La señalización planteada asegura que esta información esté disponible para cuando hay que computar y señalizar el LSP de respaldo.

Utilizando este esquema no sólo se simplifica el cómputo de los LSPs, si no que la señalización requerida para ello es sencilla, simplificando la lógica de cada nodo. Y puede resultar en un ahorro de los recursos utilizados dentro de un dominio.

Capítulo 7

Obtención de AS_PATHs disjuntos extremo a extremo

En este capítulo se va a estudiar las posibilidades de BGP para la obtención de secuencias de dominios disjuntos extremo a extremo (AS_PATHs). Se verá cuáles son sus principales limitaciones y cómo estas obstaculizan, en muchos casos, la obtención de estos AS_PATHs que permitirían el cómputo de caminos disjuntos extremo a extremo. A continuación se presentará una modificación al mecanismo de selección de rutas y a la información intercambiada en BGP que permitiría utilizar BGP como herramienta de ingeniería de tráfico para la obtención de caminos disjuntos multidominio. Se demostrará cómo estas modificaciones son válidas y ofrecen una solución siempre que el grafo de dominios sea 2-conectado. Las explicaciones se acompañarán de ejemplos. Finalmente, se realizará un resumen de lo expuesto para terminar con las conclusiones sobre el trabajo presentado.

7.1. Introducción

Cuando se quiere proteger completamente un "trunk" (o agregado de flujos) de datos entre dos nodos de la red, el camino de protección debe ser disjunto al camino principal de datos. De este modo con un sólo camino alternativo se puede proteger el tráfico ante el fallo de cualquier nodo o enlace del camino principal. Cuando el agregado de flujos atraviesa varios dominios BGP hay tres modos de obtener un camino de respaldo disjunto al camino principal.

Un primer modo es computar un par de caminos disjuntos que transcurran por los mismos dominios pero que los nodos y enlaces por los que pasan ambos caminos sólo coincidan en el nodo origen y el destino. Un segundo modo es computar un camino por una secuencia de dominios y el otro camino por otra secuencia de dominios

diferente. De esta forma cada camino se puede computar de manera independiente, ya que asegura que ambos caminos son disjuntos el hecho de que atraviesen dominios disjuntos. Por último, un tercer modo es una solución mixta, en la que parte de los caminos transcurren por distintos ASes, y por tanto intrínsecamente disjuntos, y parte del camino por los mismos ASes, y hay que computarlos de modo que lo sean en esa parte del camino.

Cuando los dominios de los que se trata son Sistemas Autónomos, o dominios BGP, el segundo y tercer modo no es posible realizarlo con la única información proporcionada por BGP. La agregación de rutas BGP, el filtrado y selección que realizan los Sistemas Autónomos de las rutas y la utilización del AS_PATH hace que no sea posible, en la mayor parte de los casos, computar dos secuencias de ASes disjuntos.

Las propuestas de esquemas de protección interdominio van destinadas, en su práctica totalidad, hacia métodos del primer modo de actuación (veánse los apartados 2.4 y 2.5), computar el camino principal y el de respaldo en la misma secuencia de dominios.

En este capítulo, en cambio, se va a proponer un sistema de cómputo interdominio del segundo modo de obtención de caminos que permita implantar diferentes esquemas de protección. Los dominios se supondrá que son Sistemas Autónomos y que utilizan BGP como protocolo de intercambio de rutas entre ellos.

Antes de presentar la propuesta se revisará brevemente el funcionamiento del intercambio de información de encaminamiento en BGP y cómo éste limita el conocimiento de la topología de dominios. En el siguiente apartado realizaremos un sucinto análisis de este mecanismo aportando algún ejemplo que nos servirá para ilustrar las limitaciones.

7.2. Funcionamiento del AS_PATH en BGP

La información de encaminamiento intercambiada en BGP es sobre la alcanzabilidad de los destinos. Esta información de alcanzabilidad viene dada, básicamente, por el AS_PATH, una lista con una cadena de Sistemas Autónomos que están "dispuestos" a llevar los paquetes a un determinado destino, el destino alcanzable por medio de ese AS_PATH y el siguiente salto para llegar al destino. Las distintas entidades BGP con esta información podrían construir un grafo de alcanzabilidad de destinos, donde los nodos intermedios son siempre Sistemas Autónomos. El mecanismo completo de intercambio de información entre *peers* BGP está definida en la RFC 1771 [85].

En cuanto al mecanismo de selección de rutas y reenvío de las mismas a los

peers vecinos se realiza en tres fases (en la figura 7.1 se presenta un esquema de este mecanismo):

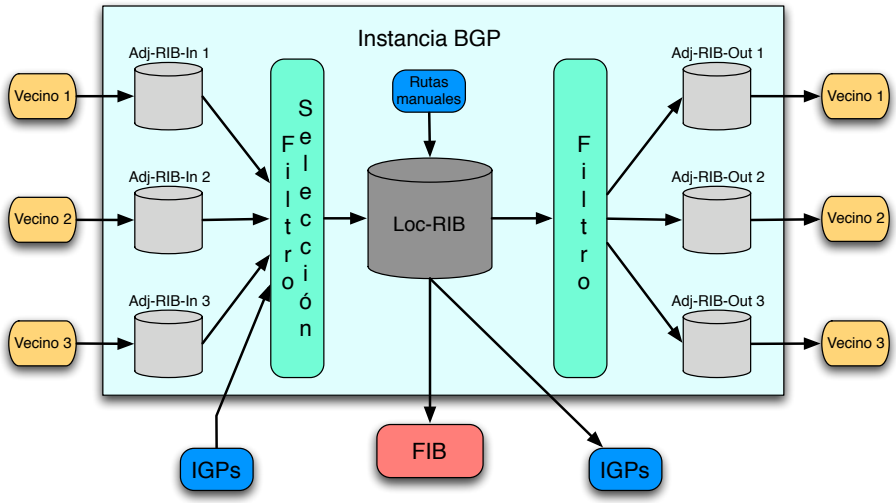


Figura 7.1: Esquema del mecanismo de selección en BGP.

1. Cuando una entidad BGP recibe un mensaje de UPDATE con una ruta de un *peer* esta se incluye en la *Adj-RIB-In*, la RIB (*Route Information Base*) con las rutas aprendidas de un determinado *peer* BGP[1]. En esta primera fase del mecanismo de selección se determina el grado de preferencia de cada una de las rutas recibidas. Cuando hay algún cambio o alguna nueva ruta en una *Adj-RIB-In* se acciona la fase dos del mecanismo de selección.

2. De entre todas las rutas disponibles para un destino, entre todas las *Adj-RIB-In*, se selecciona la considerada mejor[2]. Una vez seleccionada se incluye en la *Loc-RIB*; esta RIB contiene las rutas que se utilizarán localmente. Cuando hay algún cambio o alguna nueva ruta en la *Loc-RIB* se acciona la fase tres del mecanismo de selección.

3. Se seleccionan de entre las rutas del *Loc-RIB* cuáles se enviarán a qué *peers* BGP; para ello se incluye en las respectivas *Adj-RIB-out*, la RIB con las rutas que se envían a un determinado *peer* BGP[3], y se envían los mensajes de UPDATE correspondientes.

[1]Existe una *Adj-RIB-In* por cada *peer* BGP.

[2]Los criterios, y orden de aplicabilidad para seleccionar las rutas viene descrito en la RFC 1771. El primer criterio a tener en cuenta es el grado de preferencia de la ruta. Aquí no se describirán más detalles por carecer de interés para comprender el mecanismo general.

[3]Existe una *Adj-RIB-Out* por cada *peer* BGP.

Nótese que las rutas que se envían al resto de *peers* BGP desde una determinada entidad BGP son una selección de aquellas que están en la *Loc-RIB* y que por tanto el router utilizará localmente.

Los mensajes de UPDATE son los mensajes BGP utilizados para intercambiar información de alcanzabilidad entre los *peers* BGP. Entre otro tipo de información importante e información opcional que se puede añadir hay tres valores de especial relevancia, el AS_PATH, el NEXT_HOP y el NLRI.

El AS_PATH es la secuencia de ASes que hay que atravesar para llegar a los destinos marcados en el NLRI. Este es una lista de prefijos IP a los que se llega por medio de la ruta de ASes que se está anunciando. El NEXT_HOP es la dirección IP del router frontera del AS que envía el mensaje de UPDATE que debe ser utilizada como siguiente salto para los destinos listados en el NLRI.

Los distintos routers frontera de un Sistema Autónomo que utiliza BGP se mantienen informados de las distintas rutas recibidas por medio de IBGP, y de este modo todos los routers BGP de un AS disponen de la misma información. Por ello, excepto que se diga explícitamente lo contrario, cuando nos refiramos a un router BGP de un dominio nos referiremos a cualquiera de ellos, puesto que todos disponen de la misma información interdominio (salvo la debida a rutas específicas).

Resumiendo, un router BGP recibirá al menos una ruta para un determinado destino de cada uno de sus *peers* BGP (excepto que el router forme parte de un AS que está en el AS_PATH de la ruta elegida por parte del *peer*) y de todas estas rutas debe elegir una de ellas, añadirse como AS en el AS_PATH y anunciar la ruta a sus *peers* BGP, excepto a aquellos que formen parte del AS_PATH.

Este funcionamiento hace que la información de que dispone en las tablas RIB (*Route Information Base*) un router BGP de un AS no sea suficiente para obtener un grafo de ASes 2-conectado que le permitiese obtener AS_PATHs disjuntos a otro AS. En el siguiente apartado se utilizará un ejemplo para ilustrar el funcionamiento del intercambio de rutas en BGP y mostrar que, en general, no está disponible información suficiente para obtener dos AS_PATHs disjuntos desde un AS origen y un AS destino.

7.2.1. Ejemplo de propagación de información de alcanzabilidad en BGP y su limitación

Utilizando el mecanismo de selección de BGP se puede seguir el ejemplo propuesto en la figura 7.2. Se trata de una red de ASes interconectados, formando un grafo 2-conectado. En el ejemplo se mostrará, para las rutas destinadas al dominio *AS 4*, el estado de las diferentes tablas RIB de los routers de cada AS y se señalará

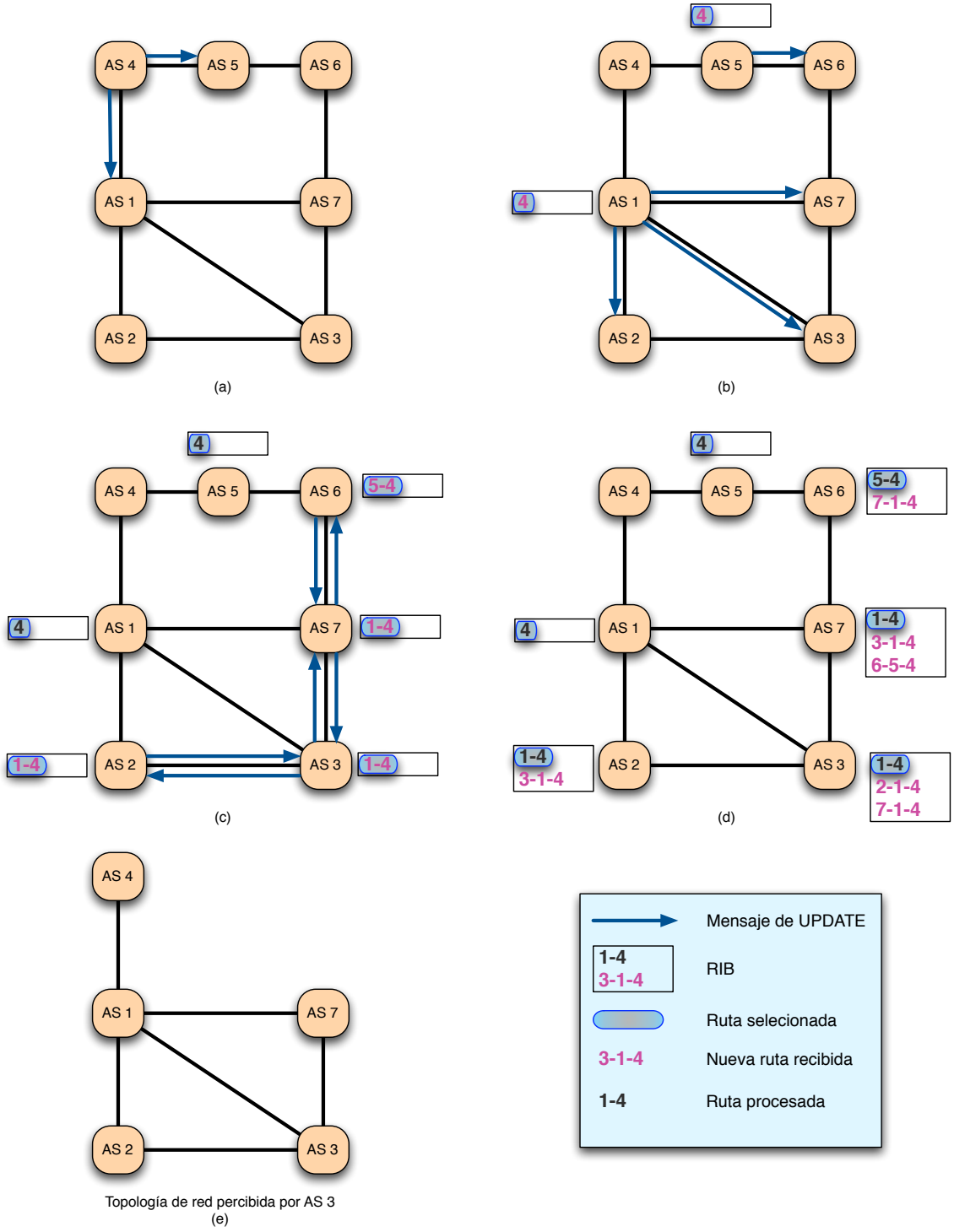


Figura 7.2: Ejemplo de propagación de rutas con BGP

cuándo se envía un mensaje de UPDATE.

En la figura se utilizan las siguientes convenciones:

- Sólo se manejan las rutas con destino *AS 4*.
- Se muestra una tabla RIB con todas las rutas manejadas.
- Si una ruta se acaba de recibir se marca en rojo. Ésta obliga a ejecutar de nuevo el mecanismo de selección.
- Las rutas antiguas están marcadas en negro.
- La ruta seleccionada, y que por tanto debe ser enviada a los *peers* BGP (excepto que forme parte de la ruta), se marca con fondo azul. Esta ruta es la que se encuentra en la *Loc-RIB*.
- Si un AS tiene que enviar un mensaje de UPDATE se indica con un flecha de color azul señalando al AS al que se le envía.
- Cada subfigura del ejemplo contiene los tres pasos del mecanismo de selección de rutas de BGP. Para cada AS se realiza lo siguiente:
  - Si en la subfigura anterior había una o más flechas terminadas en él, entonces, se incluyen las nuevas rutas en la RIB, estas se marcarán en rojo.
  - Si hay nuevas rutas marcadas en rojo en la RIB se vuelve a elegir la mejor ruta.
  - Si la ruta seleccionada ha cambiado se seleccionan los *peers* a los que debe ser enviada mediante un mensaje de UPDATE y se marca con flechas azules.
- Cuando haya que elegir entre varias rutas se elegirá la más corta (se considerará la de mayor grado de preferencia), si hay empate una de ellas al azar.

Supuesto que el dominio *AS 4* acaba de conectarse y ha establecido una sesión BGP con sus vecinos, éste empezará a anunciar a sus *peers* las redes que conoce y a las que está dispuesto a encaminar paquetes. En la figura 7.2 (a) *AS 4* está enviando un mensaje de UPDATE incluyendo los prefijos Net 4 que conoce. En la figura (b) se puede ver como los vecinos, *AS 5* y *AS 1* al recibir el mensaje de UPDATE de *AS 4* han incluido la nueva ruta en sus tablas de encaminamiento (RIB). Como la tabla de encaminamiento se ha visto modificada es necesario efectuar el paso dos del mecanismo de selección, elegir la mejor ruta al destino. En este caso como sólo hay una ruta se selecciona esta directamente. Debido a que la nueva ruta seleccionada

es diferente a la que se había seleccionado previamente (no había ninguna) debe anunciarse la existencia de esta nueva ruta a los ASes vecinos, a los que se les envían los mensajes de UPDATE. Al llegar los mensajes de UPDATE a los distintos ASes, figura (c), estos incluyen las nuevas rutas en las RIBs, se procesan y se selecciona la mejor.

Al llegar estos mensajes a los ASes, estos incluyen las nuevas rutas en las RIBs, figura (d). Ninguna de estas nuevas rutas ha sido seleccionada como mejor que la que se tenía, por lo que no hay ninguna ruta actualizada y por tanto ningún AS tiene que enviar un mensaje de UPDATE. Se llega a la estabilidad en la red, teniendo cada AS una ruta para llegar a *AS 4*.

Puede observarse cómo al final del proceso todos los dominios tienen una ruta seleccionada en las RIB, y por tanto también en las FIB (*Forwarding Information Base*), para alcanzar el destino *AS 4*. Este es el cometido de BGP: proporcionar un camino de ASes para alcanzar un determinado destino. Sin embargo, también puede verse que en las tablas RIB no hay información suficiente para obtener un camino alternativo disjunto al destino, sólo disponen de ella algunos de los ASes. Es el caso de *AS 6* y *AS 7* en la figura. Por ejemplo, *AS 6* tiene en su FIB que para llegar a *AS 4* debe ir por *AS 5*, pero además dispone en sus tablas RIB de la ruta *AS 7-AS 1*. Esta información no está disponible en las tablas de reenvío, pero una herramienta de Ingeniería de Tráfico como MPLS podría utilizar esta información para crear un LSP de respaldo pasando por estos dominios.

El principal problema es que en la mayoría de los casos la información necesaria no está disponible de manera global, como se requeriría para realizar conexiones MPLS multidominio. El motivo es precisamente el diseño de los protocolos de encaminamiento interdominio, orientados a la escalabilidad mediante agregación (con pérdida) de la información de alcanzabilidad. Ejemplo de esto puede verse en la figura (e), en la que se muestra la topología de red percibida por *AS 3* mediante BGP.

En la siguiente sección presentaremos un mecanismo que nos permitirá disponer de esta información en todos los dominios.

7.3. Porpuesta de extensión de BGP para poder obtener AS_PATHs disjuntos entre ASes

En este apartado se va a proponer una modificación que puede realizarse a BGP para que todos los ASes dispongan de información suficiente como para poder obtener dos secuencias de AS disjuntas para alcanzar un destino. Es decir, que cualquier Sistema Autónomo sea capaz de obtener un ciclo de ASes que contenga al destino y a sí mismo. Todo esto sin alterar el mecanismo de selección de la ruta principal de

BGP.

Primero se describirá en que consiste la modificación a realizar, se expondrá un ejemplo de funcionamiento, para a continuación demostrar la validez funcional de la propuesta, y terminar con un análisis de escalabilidad del mecanismo propuesto.

7.3.1. Propuesta de modificación de BGP

Para el mecanismo que se va a plantear son necesarias tres nuevas RIBs, la *Adj-RIB-Disj-In*, la *Loc-RIB-Disj* y la *Adj-RIB-Disj-Out*. Si bien, se pueden utilizar las RIB originales de BGP [85] añadiendo un *flag* que indique si la ruta es principal o secundaria (disjunta).

El mecanismo para la obtención, cálculo y reenvío de las rutas principales no se ve afectado ni modificado en modo alguno, manteniendo los tiempos de convergencia actuales, así como el número de mensajes intercambiados o el tamaño de las tablas. La idea es establecer un mecanismo en *segundo plano* para la obtención de rutas disjuntas a las principales, llamadas *rutas secundarias*. Para ello también se utiliza un mecanismo en tres fases similar al que se utiliza en las rutas principales (en la figura 7.3 se presenta un esquema de este nuevo mecanismo):

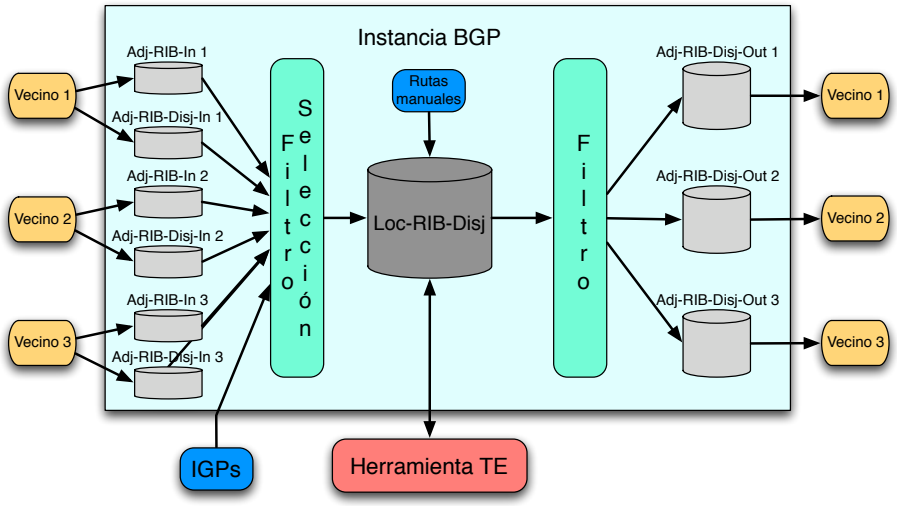


Figura 7.3: Esquema del mecanismo de selección de rutas BGP disjuntas propuesto.

1. Cuando una entidad BGP recibe un mensaje de UPDATE con una ruta secundaria de un *peer* incluye esta ruta en la *Adj-RIB-Disj-In*, la RIB con las rutas secundarias aprendidas de un determinado *peer* BGP. En esta primera fase del mecanismo de selección se determina el grado de preferencia de cada una

de estas rutas recibidas. Cuando hay un cambio o alguna nueva ruta en una *Adj-RIB-In* o en una *Adj-RIB-Disj-In* se acciona la fase dos de este mecanismo. También se activa cuando hay un cambio en la *Loc-RIB*, ya que indica un cambio de ruta principal y esto puede requerir un cambio en la ruta secundaria.

2. De entre todas las rutas disponibles para un destino (tanto principales, incluidas en *Adj-RIB-In*, como secundarias, incluidas en *Adj-RIB-Disj-In*) se selecciona, de entre las disjuntas a la ruta principal, la mejor. Si se ha seleccionado alguna se incluye en la *Loc-RIB-Disj*; esta RIB contiene las rutas secundarias que se utilizarán localmente. Cuando hay algún cambio o una nueva ruta en la *Loc-RIB-Disj* se acciona la fase tres de este mecanismo de selección.

   Si no se ha podido seleccionar una ruta alternativa disjunta se debe comprobar si existe una ruta entre las principales para la que sí es posible obtener una ruta secundaria disjunta. En tal caso, se debe disminuir el grado de preferencia de la ruta principal actual de modo que sea elegida la otra ruta principal, para la que sí existe una ruta secundaria disjunta. Este mecanismo evita que la selección de la ruta principal sea una "trampa" para la que no hay ruta disjunta (aún existiendo dos rutas disjuntas entre sí). Si se recibe una nueva ruta secundaria se debe comprobar si la ruta anterior es válida para restituir su grado de preferencia.

3. Se seleccionan de entre las rutas del *Loc-RIB-Disj* cuáles se enviarán a qué *peers* BGP, se incluyen en las respectivas *Adj-RIB-Disj-out* y se envían los mensajes de UPDATE con las rutas secundarias correspondientes.

Una ruta secundaria no puede ser elegida como ruta principal, puesto que el siguiente AS no encaminaría bien los paquetes, debido a que al tratarse de una ruta secundaria no tiene por qué estar en uso. Sin embargo, una ruta principal sí puede ser seleccionada como ruta secundaria, ya que se eligen las rutas disjuntas a la principal tanto de *Adj-RIB-In* como de *Adj-RIB-Disj-In*.

Si ha habido un cambio en las tablas y se selecciona una ruta principal, que previamente había sido elegida como secundaria, se activa el paso dos del mecanismo, que obliga a elegir otra ruta secundaria, ya que esta debe ser disjunta a la recién seleccionada como ruta principal.

En el siguiente apartado se mostrará un ejemplo de utilización del nuevo mecanismo sobre la red del ejemplo anterior.

7.3.2. Ejemplo de funcionamiento

Sobre la misma red que se utilizó en el ejemplo anterior se va mostrar el funcionamiento de este nuevo mecanismo. Todo el ejemplo se muestra en la figura 7.4.

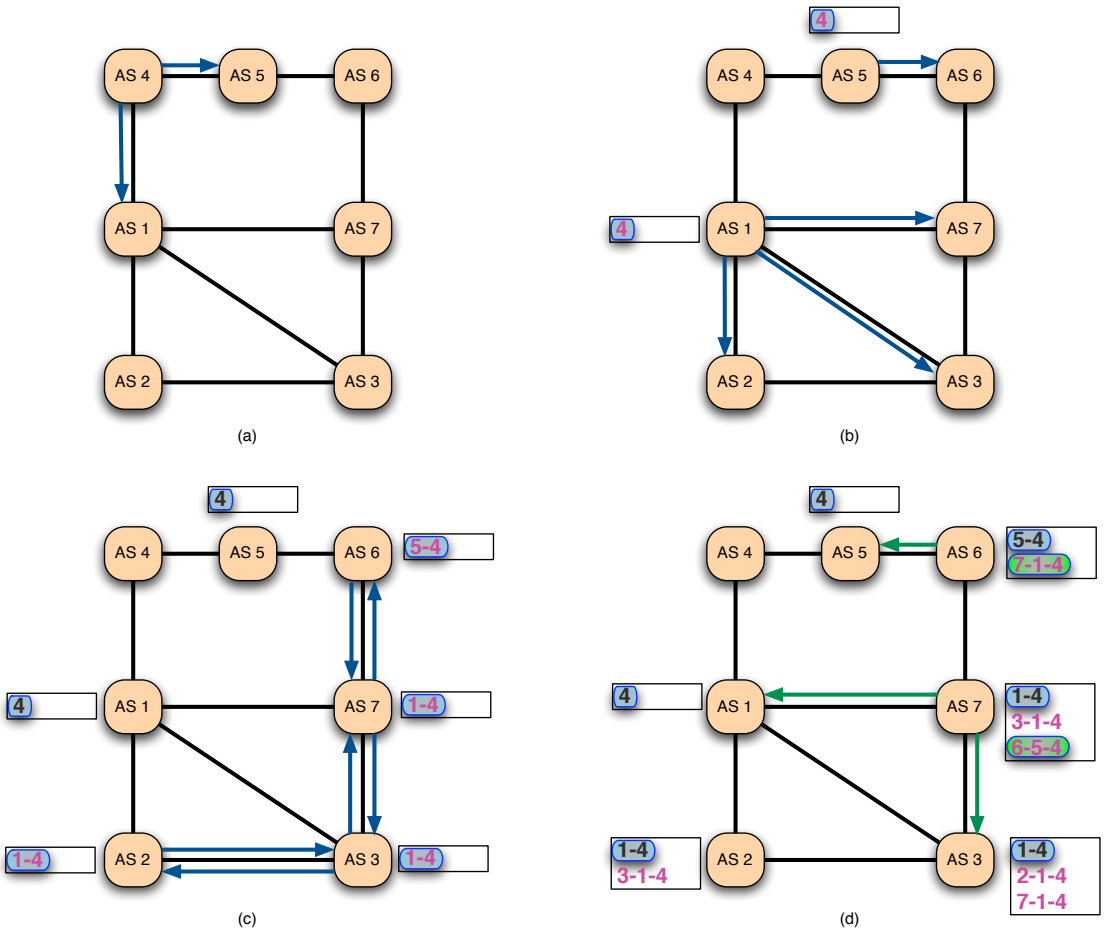
Para esta figura se utilizan ademas de las convenciones utilizadas en el ejemplo anterior las siguientes:

- Los mensajes de UPDATE de rutas secundarias se marcan en verde.
- Las rutas secundarias seleccionadas se marcan con fondo verde.
- Las rutas que se han recibido como rutas secundarias se marcan entre "*".
  - De color rojo para las recién recibidas.
  - De color negro para las antiguas.
- Cada subfigura del ejemplo contiene además de los tres pasos del mecanismo de selección de rutas de BGP los tres pasos del mecanismo de selección de rutas secundarias. Para cada AS se realiza lo siguiente:
  - Si en la subfigura anterior había una o más flechas terminadas en él, entonces: se incluyen las nuevas rutas en la RIB, se marcan en rojo, indicando si son principales (sin "*") o secundarias (con "*").
  - Si hay nuevas rutas marcadas en rojo en la RIB se vuelve a elegir la mejor ruta (principal y/o secundaria, según corresponda).
  - Si la ruta seleccionada (principal o secundaria) ha cambiado se seleccionan los *peers* a los que debe ser enviada mediante un mensaje de UPDATE. En la figura se marca con flechas azules (principal) o verdes (secundaria).

Como puede verse en la figura, los 4 primeros pasos, hasta el (d), son los mismos que en el ejemplo 7.2, ya que hasta este momento no ha sido posible calcular ninguna ruta disjunta a las principales en ningún nodo. Las rutas principales obtenidas en los distintos nodos han seguido el mismo proceso y son las mismas. Es en ese paso, el (d), cuando varios dominios, *AS 6* y *AS 7*, tienen en la *Adj-RIB-In* rutas disjuntas (rutas sin "*") a las rutas principales seleccionadas (rutas sin "*" con fondo azul).

A partir de este momento no hay ningún cambio en las rutas principales debido a que no se añade ninguna nueva ruta en las *Adj-RIB-In*, ni se modifica el grado de preferencia a ninguna ruta, lo que hace que no se dispare el mecanismo de selección de rutas de BGP (paso 2).

Volviendo al ejemplo, en la figura (d), la ruta recibida por *AS 6* desde *AS 7*, 7-1-4, es disjunta a la seleccionada previamente como principal, 5-4, y se marca como ruta secundaria (fondo verde). De igual modo ocurre en *AS 7*, siendo elegida la ruta 6-5-4 como ruta secundaria. Estas nuevas rutas secundarias seleccionadas provocan el


AS 4
AS 5
AS 6
AS 1
AS 7
AS 2
AS 3
(a)
4
(b)
5-4
1-4
(c)
7-1-4
3-1-4
6-5-4
2-1-4
(d)

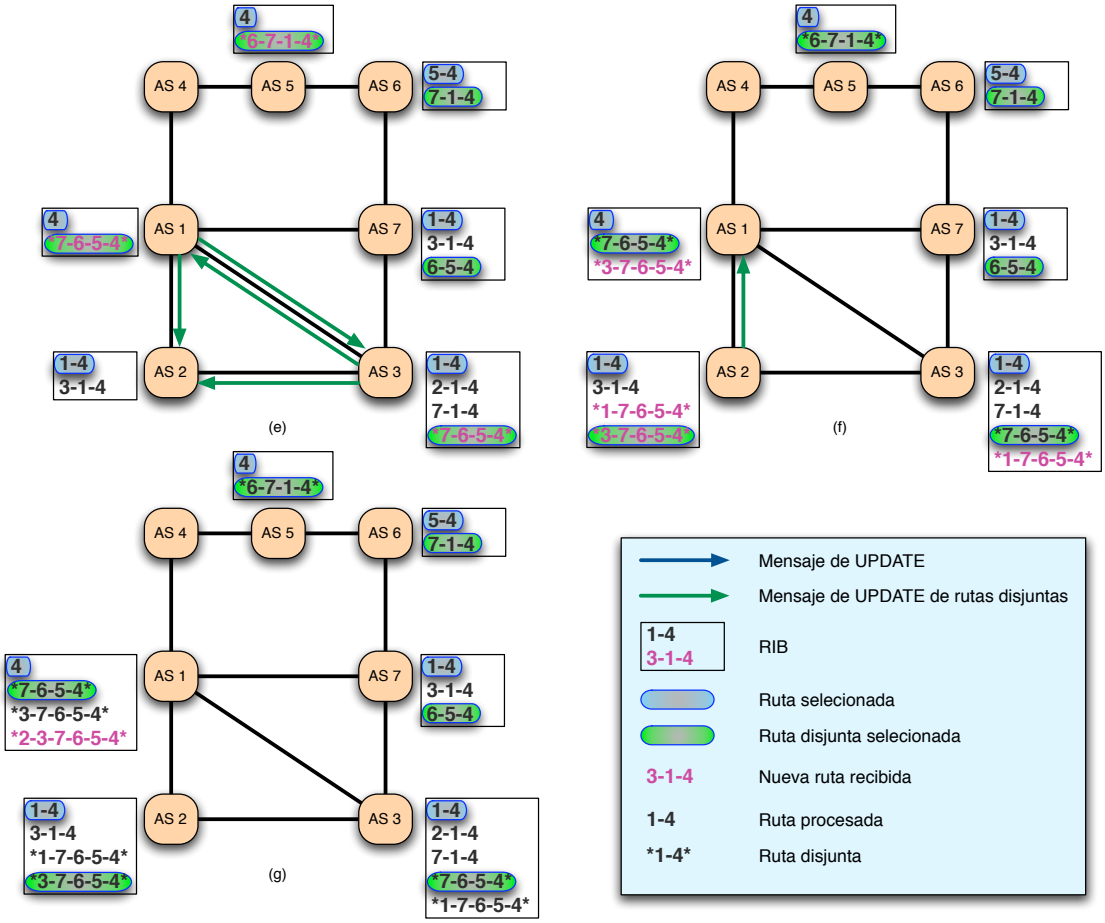


Figura 7.4: Ejemplo de propagación de rutas con la modificación de BGP introducida

envío de nuevos mensajes de UPDATE de rutas secundarias a los vecinos (marcados con flechas de color verde).

Una vez enviados los nuevos mensajes de UPDATE, figura (e), las rutas nuevas se guardan en la *Adj-RIB-Disj-In* correspondiente (rutas con "*"). Después el paso 2 del mecanismo de selección determina las rutas secundarias más adecuadas en todos aquellos AS en los que se tiene una ruta secundaria nueva. Todos aquellos nodos que seleccionan una ruta y esta es distinta a la selección anterior deben ejecutar el paso tres del mecanismo de selección, enviar los mensajes de UPDATE con las nuevas rutas secundarias a los vecinos (flechas de color verde).

Siguiendo este proceso en los pasos (f) y (g) llegamos al estado final, en el que todos los AS han seleccionado una ruta principal y una secundaria disjunta y no ha habido ningún cambio respecto a la situación anterior.

En este estado final puede verse que todos los ASes disponen de las mismas rutas principales que en el ejemplo anterior, conseguida además en el mismo momento, y que todos disponen de una ruta alternativa, disjunta a la principal, para llegar al destino tratado, el *AS 4*. Por ejemplo, *AS 3* dispone de la ruta 1-4 para llegar al dominio *AS 4* y dispone de la ruta disjunta alternativa 7-6-5-4. Lo mismo ocurre con el resto de ASes.

En el siguiente apartado se demostrará que utilizando este nuevo mecanismo de selección e intercambio de rutas todos los dominios disponen de una ruta alternativa y disjunta para llegar a los distintos destinos. También se discutirá la utilidad de esta información.

7.3.3. Demostración

En este apartado se va a demostrar que utilizando el mecanismo propuesto todos los nodos de la red, si esta es 2-conectada, dispondrán de dos rutas, una principal y otra disjunta. Se irán proponiendo y demostrando teoremas que servirán para la demostración del teorema final, el teorema 7.4.

Hay una serie de características del intercambio de rutas en BGP (algunas de ellas comunes a todos los protocolos de encaminamiento) que es preciso introducir:

Proposición 7.1. ***Principio de optimalidad***. *Si el dominio* AS O *tiene una ruta con destino* AS D *seleccionada y pasa por el dominio* AS I*, entonces la ruta entre* AS I *y* AS D *es común para* ASO *y* AS I.

Demostración. AS I sólo anuncia el AS_PATH, AS_PATH_{ID}, hacia el destino *AS D* que él mismo utiliza. De los AS_PATH recibidos sólo se utiliza uno, en ningún caso se combinan varios para obtener el que se utilizará. Al AS_PATH elegido se le

añade información, e.g. el propio número AS, en ningún caso se modifica o elimina información existente. El AS_PATH elegido por *AS I*, AS_PATH_{ID}, es la única ruta que *AS I* propagará con destino a *AS D* y el resto de ASes sólo le añadirán nuevos números AS. Cualquier AS_PATH destinado a *AS D* que contenga a *AS I* es porque contiene la ruta AS_PATH_{ID}. Por tanto, cualquier dominio *AS O* que tenga un AS_PATH destinado a *AS D* y pase por *AS I* contiene AS_PATH_{ID}, por lo que comparten todo el camino desde *AS I* hasta *AS D*. □

La topología de ASes es 2-conectada; una propiedad de este tipo de redes es que siempre es posible obtener dos caminos disjuntos desde un nodo X de la red a otro nodo Y cualesquiera de ella (veáse el apéndice A). A estos caminos los llamaremos AS_PATH_{XY} y $AS\_PATH\_DISJ_{XY}$ respectivamente.

Teorema 7.1. *Supuestos dos ASes,* AS A *y* AS B*, conectados a un tercero,* AS O *de modo que existen los AS_PATHs;* AS_PATH_{AD}, AS_PATH_{BD} *y* $AS\_PATH\_DISJ_{BD}$ *con destino* AS D*. Si el camino principal de* AS O *hacia* AS D *es vía* AS A*, por lo que* $AS\_PATH_{OD} = AS\ O - AS\_PATH_{AD}$*, entonces, existe un* $AS\_PATH\_DISJ_{OD}$ *disjunto a* AS_PATH_{OD} *tal que* $AS\_PATH\_DISJ_{OD} = AS\ O - AS\_PATH_{BD}$ *o* $AS\_PATH\_DISJ_{OD} = AS\ O - AS\_PATH\_DISJ_{BD}$.

Demostración. Como puede observarse en la figura 7.5 el camino disjunto al camino principal hasta *AS D* tiene que ser alguno provisto por *AS B*, ya que la ruta principal de *AS O* pasa por *AS A* y la ruta secundaria no puede pasar por *AS A*.

Cualquiera de los dos caminos, AS_PATH_{BD} o $AS\_PATH\_DISJ_{BD}$, que no corte a AS_PATH_{AD} es válido como solución. No es posible que los dos corten con AS_PATH_{AD} porque si no, según la proposición 7.1, ambos caminos compartirían el final de sus trayectos entre sí y con AS_PATH_{AD}, pero AS_PATH_{BD} y $AS\_PATH\_DISJ_{BD}$ son disjuntos, por lo que, al menos uno de ellos es disjunto a AS_PATH_{AD} y, por tanto, solución. □

Corolario 7.1.1. *Si tanto* AS A *como* AS B *disponen de dos AS_PATHs disjuntos a* AS D*, entonces* AS O *podrá obtener siempre dos AS_PATHs disjuntos a* AS D

Demostración. Independientemente de qué AS, *AS A* o *AS B*, se seleccione para el camino principal hacia *AS D*, el otro AS dispone de dos AS_PATHs disjuntos a *AS D*, con lo que es posible aplicar el teorema 7.1. □

Corolario 7.1.2. *Si se dispone de dos parejas de AS_PATHs disjuntos desde un AS (*AS O*) a otros dos ASes (*AS A *y* AS B*) entonces es posible obtener un AS_PATH entre* AS O *y* AS A *disjunto a otro AS_PATH entre* AS O *y* AS B.

Es fácil ver que una red 2-conectada está formada por ciclos unidos, precisamente una de las características de estas redes es poder computar caminos disjuntos

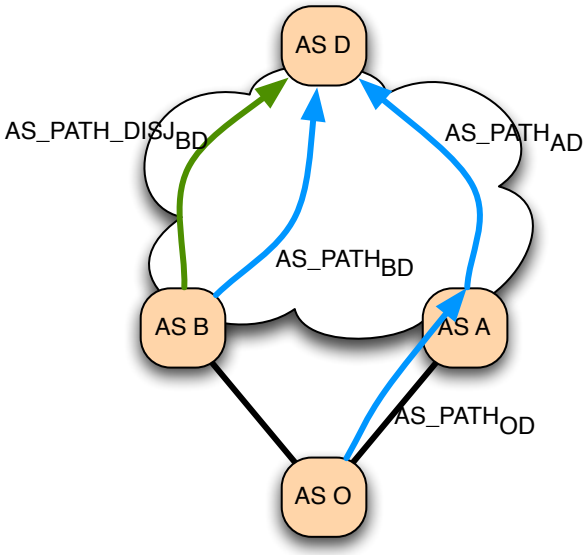


Figura 7.5: Propagación de rutas secundarias.

desde un nodo de la red a otro. Estos dos caminos forman siempre un ciclo, véase la figura 7.6.

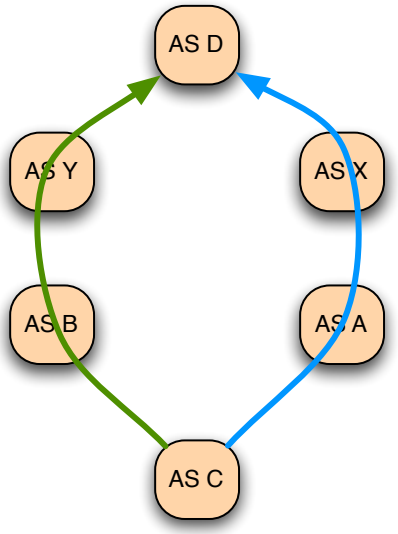


Figura 7.6: Caminos disjuntos en una red 2-conectada forman un ciclo.

Lema 7.1. *Utilizando el mecanismo de selección de rutas propuesto en la sección 7.3.1 se cumple el teorema 7.1.*

Demostración. La única forma en la que un mecanismo de selección no cumpliría con el teorema 7.1 sería que dicho mecanismo eligiese como AS_PATH_{OD} a $AS\ O - AS\_PATH_{BD}$ y que, además, AS_PATH_{AD} no fuese disjunto a AS_PATH_{BD}, de modo que no fuese posible seleccionar un $AS\_PATH\_DISJ_{OD}$ válido (veáse la figura 7.5). Si esto ocurriese, en el mecanismo propuesto aquí, al AS_PATH elegido como principal (AS_PATH_{BD}) se le baja el grado de preferencia, lo que hace ejecutar de nuevo el mecanismo de selección del AS_PATH principal, de modo que en esta ocasión será elegido el AS_PATH_{AD}, cumpliéndose así el teorema 7.1. □

Lema 7.2. *En una red 2-conectada de Sistemas Autónomos existen dos nodos adyacentes de cada ciclo simple*[4] *que reciben al menos dos rutas BGP para llegar a un destino y éstas son disjuntas. Esto es cumple para cualquier destino del ciclo*

Demostración. Si un determinado AS_PATH, *ASP 1*, a un destino, *AS D*, avanza por una secuencia de ASes dispuestos en ciclo este dejará de avanzar cuando en un AS no sea seleccionado ese AS_PATH. El nodo que no seleccione dicho AS_PATH, *AS X*, es porque ha recibido otro AS_PATH, *ASP 2*, "mejor" a ese destino, por el otro lado del ciclo. El AS adyacente, *AS Y*, que fue el último en seleccionar *ASP 1* recibirá de *AS X* el *ASP 2*, por ser la ruta que *AS X* ha seleccionado para *AS D* y no estar *AS Y* en ella, ya que el AS_PATH destinado a *AS D* que contiene a *AS Y* es *ASP 1* y no *ASP 2*.

AS X y *AS Y* disponen ambos de los AS_PATHs *ASP 1* y *ASP 2*, que además son disjuntos, por contener cada uno de ellos secciones del ciclo distintas. En la figura 7.7 se puede ver gráficamente. □

Teorema 7.2. *Utilizando el mecanismo de selección de rutas propuesto en la sección 7.3.1 todos los nodos de un ciclo simple obtienen un AS_PATH y un AS_PATH_DISJ disjuntos a cualquier otro nodo del ciclo.*

Demostración. Según el lema 7.2 hay dos nodos con dos rutas disjuntas en sus RIBs para un determinado destino. Una de ellas será la ruta principal, la otra será seleccionada por el mecanismo de selección de rutas como la ruta secundaria. Uno de estos dominios (*AS X* en la figura 7.7) tiene como dominio adyacente el dominio del que aprendió la ruta principal (*AS F*). Entonces, existe un dominio (*AS F*) adyacente a otro dominio (*AS X*) del que no aprendió la ruta principal y que dispone de dos rutas disjuntas. Según el teorema 7.1 aquel dominio (*AS F*) es capaz de obtener la ruta secundaria disjunta de la principal hacia el destino.

Este dominio, con dos rutas disjuntas (*AS F*), ofrece la misma posibilidad al siguiente dominio del ciclo. Así hasta llegar al destino. En el otro semiciclo las rutas

[4]Con ciclo simple se quiere recalcar que el conjunto de nodos forman un único ciclo, es decir, que a cada nodo sólo se le consideran dos enlaces.

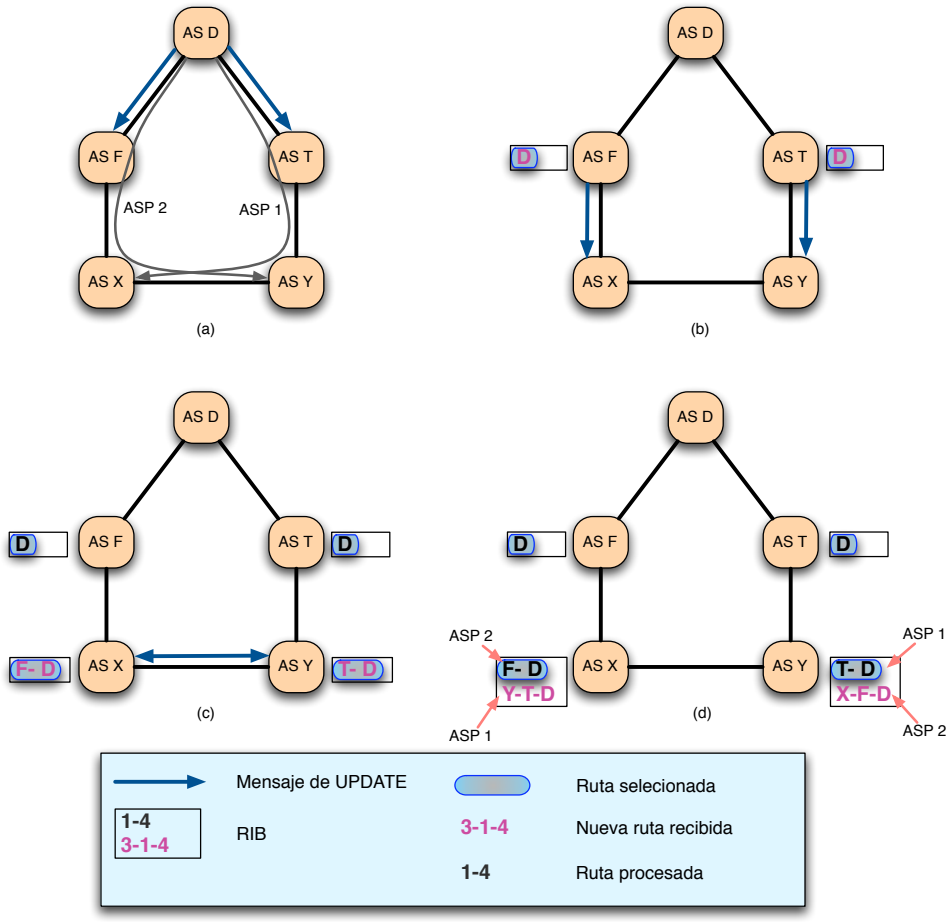


Figura 7.7: Al menos dos nodos BGP de un ciclo tienen al menos 2 rutas a un destino concreto del ciclo.

disjuntas avanzan del mismo modo desde el otro dominio que disponía de dos rutas disjuntas (*AS Y*). □

Teorema 7.3. *Si dos o más nodos adyacentes de un ciclo simple tienen un AS_PATH y un AS_PATH_DISJ disjuntos a un determinado destino X (no necesariamente perteneciente al ciclo), entonces utilizando el mecanismo de selección propuesto en la sección 7.3.1 todos los nodos del ciclo obtendrán un AS_PATH y un AS_PATH_DISJ disjuntos al destino X.*

Demostración. Sabemos, por el teorema 7.2, que dos ASes adyacentes, *AS A* y *AS B* (ejemplo de la figura 7.8), tienen dos rutas disjuntas a *AS D* y que *AS O* dispone también de dos rutas disjuntas a *AS A* y *AS B*. Por tanto, según el corolario 7.1.2, *AS D* puede obtener dos caminos disjuntos hasta *AS A* y *AS B*.

Es posible considerar a *AS A* y *AS B* como un único AS, que inyecta al ciclo dos pares de rutas disjuntas hacia *AS D*. Siguiendo el mismo razonamiento que en el lema 7.2, este par de rutas disjuntas se propagarán por el ciclo. Cuando un AS recibe los dos pares de rutas, de entre las primarias debe elegir la que utilizará como primaria. Si elige la del *AS A* la ruta secundaria la seleccionará de entre las dos que pasan por *AS B* (corolario 7.1.1). □

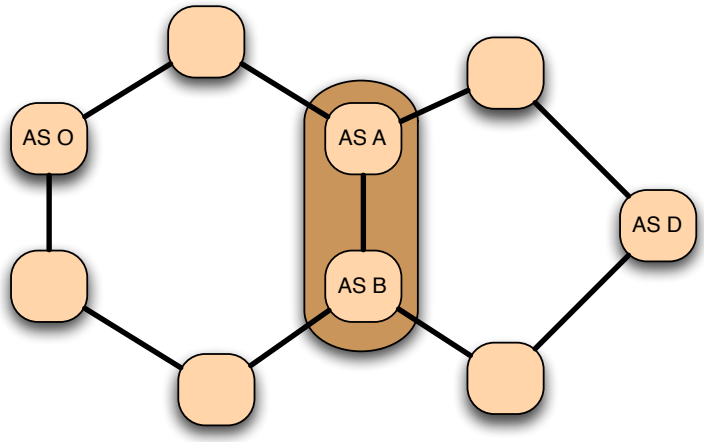


Figura 7.8: Dos nodos en un ciclo con dos caminos disjuntos a un destino cada uno permiten calcular dos caminos disjuntos al resto de nodos.

Teorema 7.4. *Utilizando el mecanismo de selección propuesto en la sección 7.3.1 todos los dominios de una red de Sistemas Autónomos 2-conectada obtienen un AS_PATH y un AS_PATH_DISJ disjuntos a cada destino de la red.*

Demostración. Una red 2-conectada está formada por al menos un ciclo simple. Puede estar formada por más de un ciclo, en cuyo caso los ciclos se unen entre sí para formar la red. Para que un ciclo forme parte de la red tiene que tener en común al menos dos nodos con otro ciclo de la red, sino dejaría de ser una red 2-conectada. Por ejemplo, en la red de la figura 7.9 los ciclos A y B no forman una red 2-conectada porque el nodo común Y es el único nodo que interconecta ambos ciclos, con lo que no es posible encontrar dos caminos disjuntos entre un nodo del ciclo A, $AS\ S$, y otro del ciclo B[5], $AS\ Z$, ya que ambos deberían de pasar por Y.

Según el teorema 7.2, todos los nodos de los ciclos simples que contienen el AS destino obtienen 2 rutas BGP disjuntas al AS destino. Cualquier ciclo adyacente a uno de estos comparte dos o más nodos contiguos, sino no sería una red 2-conectada. Entonces, un ciclo adyacente a otro que contiene el dominio destino tiene dos o más nodos consecutivos con dos rutas disjuntas al dominio destino, por el teorema 7.3 todos los nodos de ese ciclo pueden obtener dos rutas disjuntas al destino. Lo mismo ocurrirá con los ciclos adyacentes a este, y así de manera recursiva con toda la red.

Explicado para un destino, lo mismo es válido para todos los posibles destinos de la red, propagando las rutas desde su propio ciclo hacia los demás. □

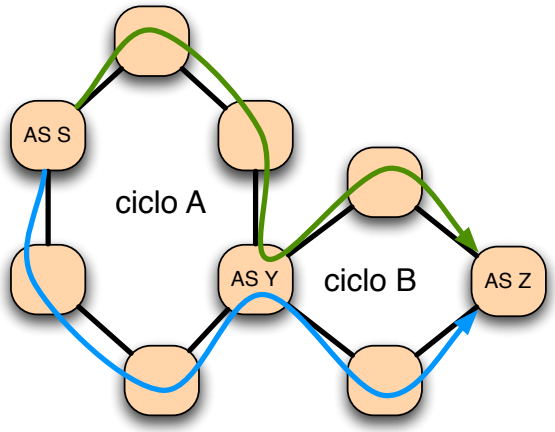


Figura 7.9: Una red con dos ciclos y un sólo nodo común entre los ciclos no es 2-conectada

En el siguiente apartado se va a estudiar un ejemplo de uso del esquema de encaminamiento propuesto y cómo este puede ser útil en la creación de caminos de respaldo de conmutación rápida utilizando MPLS.

[5]Recuerdese que una de las principales características de las redes 2-conectadas es que es posible calcular dos caminos disjuntos entre dos nodos cualesquiera de la red.

7.3.4. Análisis del mecanismo propuesto mediante un ejemplo de uso

En este apartado se va a examinar en detalle un ejemplo más complejo de utilización del mecanismo propuesto. Una vez obtenidas las rutas por todos los Sistemas Autónomos se verá cómo utilizar estas para realizar protección interdominio y cómo sería el funcionamiento en caso de fallo. El ejemplo es el mostrado en la figura 7.10. Se ha elegido este ejemplo por formar parte de la familia de grafos de red en los que BGP tiene una mala convergencia, según se ha estudiado en la literatura [6][86][87].

7.3.4.1. Ejemplo

En este ejemplo se utilizan los mismos convencionalismos generales que en la figura 7.4. Además existen los siguientes particulares para esta figura:

- *AS 3* no forma parte de los dominios que tienen acuerdos para reenviarse rutas secundarias.
  - Sus *peers* no le enviarán UPDATEs con rutas secundarias.
  - *AS 3* no seleccionará una ruta de sus RIBs como ruta secundaria para ningún destino para enviársela a sus *peers* BGP.
- *AS 1*, debido a las preferencias locales, prefiere la ruta por *AS 3* antes que la ruta por *AS 6* para alcanzar *AS 7*.
  - Esto provoca que en la iteración 4, figura (d), *AS 1* cambie de ruta principal, enviando un UPDATE con la nueva ruta y un WITHDRAWAL con la antigua, que obliga a eliminar de las RIBs a los *peers* dicha ruta.

Merece especial mención lo que le ocurre a *AS 2* con su ruta secundaria (lo mismo pasa con *AS 4*). En el paso (d), *AS 2* ha seleccionado de entre las rutas principales a 1-6-7 como su ruta secundaria y envía UPDATEs con esta información a sus *peers*. Sin embargo cuando *AS 1* recibe la ruta que pasa por *AS 3* este la prefiere a 6-7 y la cambia, obligando a eliminarla de las tablas al resto de ASes, incluido *AS 2*. En el mismo momento, *AS 1* selecciona dicha ruta como ruta secundaria, con lo que junto al WITHDRAWAL de 1-6-7 como ruta principal envía un UPDATE con la misma ruta como ruta secundaria. *AS 2*, en el paso (e), selecciona esta misma ruta como ruta secundaria y por tanto no envía nada.

En una implementación real, y dependiendo de los tiempos de recepción de los UPDATEs y WITHDRAWALs, existirían pasos intermedios de intercambio de rutas

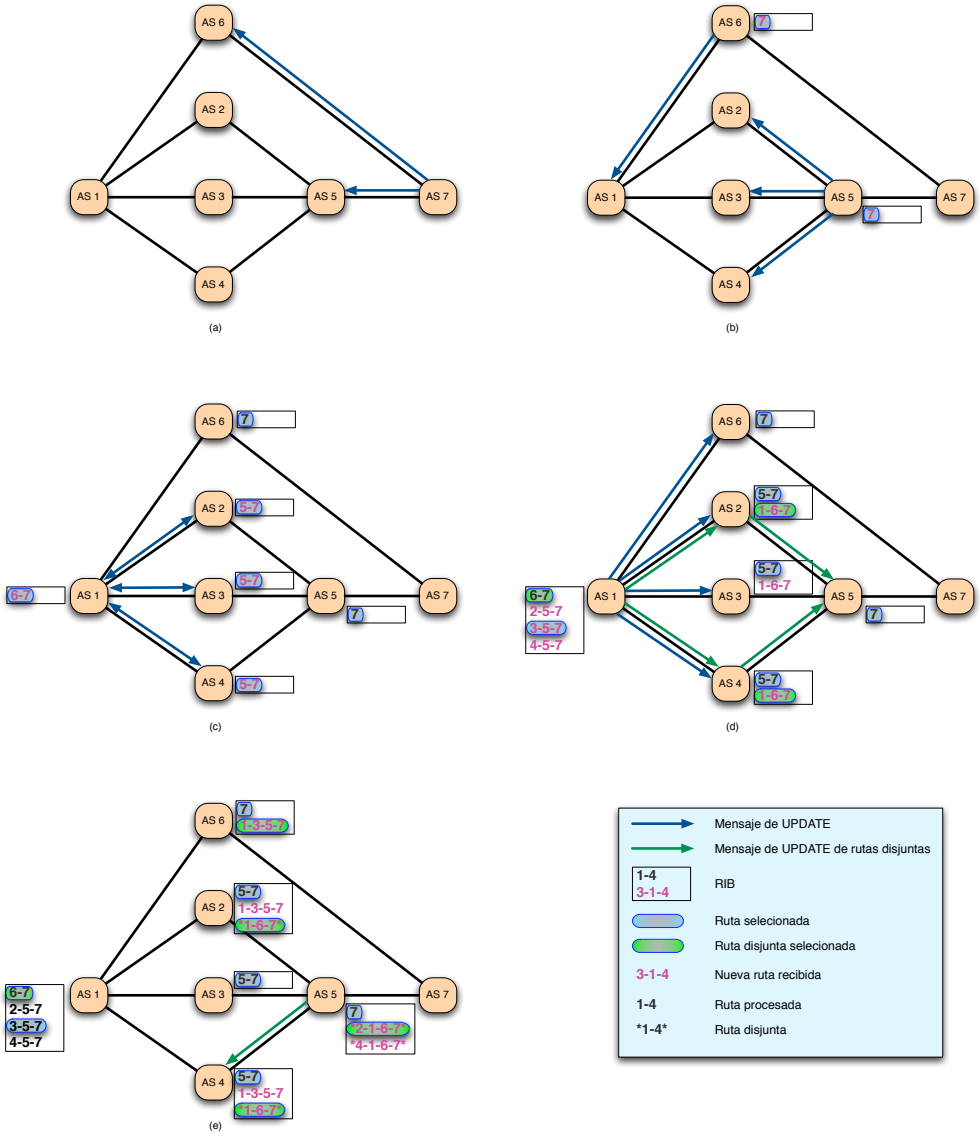


Figura 7.10: Rutas alternativas con BGP en topologías de convergencia lenta.

hasta llegar a esta situación, puesto que la ruta, aunque es la misma, es de otra RIB y provocaría los correspondientes mensajes de WITHDRAWAL y UPDATE. Por claridad en el esquema del ejemplo se han omitido estos pasos intermedios. Fíjese que, a pesar de esto, los mensajes correspondientes a las rutas principales no varían respecto al caso de utilizar el mecanismo de BGP tradicional, primero un UPDATE, y a continuación el WITHDRAWAL y UPDATE para rectificar la ruta.

Teniendo en cuenta sólo el mecanismo tradicional de BGP, un fallo en el enlace *AS 5-AS 7* puede provocar un proceso de "cuenta hasta infinito" [86] que sume a *AS 1* en creer que puede seguir alcanzando a *AS 7* por *AS 2*, *AS 3* o *AS 4* alternativamente. Al final del proceso de convergencia *AS 1* podría enviar el tráfico por *AS 6* sin pérdidas.

Sin embargo, como puede verse en la figura (e), utilizando el esquema de encaminamiento propuesto, *AS 1* tiene conocimiento de una ruta alternativa a *AS 7* en todo momento, con lo que podría utilizarla para reencaminar el tráfico aunque el enlace entre *AS 5-AS 7* siga sin funcionar. Una vez solucionado el fallo o cuando la estabilidad en las rutas BGP ofrezcan un camino alternativo, *AS 1* podría enviar su tráfico por la ruta indicada por BGP.

Si bien, para poder recuperarse rápidamente del fallo en el enlace *AS 5-AS 7* no basta con conocer esta ruta adicional, es necesario poder actuar y hacerlo a tiempo. Utilizando sólo IP/BGP no es trivial implementar un mecanismo que solvente la situación mientras BGP converge. El funcionamiento de IP/BGP hace que el *tiempo de notificación* para *AS 1*, tiempo que transcurre desde que ocurre el fallo hasta que *AS 1* detecta el fallo, es elevado. De hecho, *AS 1* no detecta ningún fallo, sólo notará que han cambiado algunas rutas hacia ciertos destinos, pero no puede saber el motivo real, con lo cuál, aunque pudiese actuar, no sabría a que reaccionar, ni que rutas alternativas puede o debe utilizar, ya que no tiene conocimiento de que enlace o dominio ha fallado. Es necesario una herramienta de Ingeniería de Tráfico, i.e. MPLS, para poder saber a qué reaccionar y cómo reaccionar, también para poder utilizar las rutas alternativas obtenidas con el mecanismo de selección de rutas.

7.3.4.2. Recuperación rápida en caso de fallo

Supuesto que *AS 1* ha completado el aprendizaje de rutas BGP, y que dispone de las rutas principal y secundaria a *AS 7* se va a mostrar cómo *AS 1* mediante MPLS, y con ayuda de las rutas secundarias, puede realizar un *fast reroute* del tráfico destinado a *AS 7*.

En *AS 1* para transportar el "trunk" que se quiere proteger destinado a *AS 7* puede construirse un LSP interdominio utilizando la ruta habitual de BGP, 3-5-7. Para ello, basta con señalizar un LSP utilizando RSVP-TE con un objeto ERO con los dominios *AS3*, *AS 5* y *AS 7* como "strict hops" [88]. De este modo se creará

un LSP extremo a extremo interdominio desde el router R_{13} hasta el router R_{75}, figura 7.11, que atraviesa los dominios *AS 3* y *AS 5*. El tráfico del dominio *AS 1* destinado a *AS 7* alcanzará el router R_{13} y este lo introducirá por el túnel *LSP 1* hasta *AS 7*. Es decisión del dominio *AS 1* si se utilizan LSPs intradominio para alcanzar R_{13} o reenvío IP convencional.

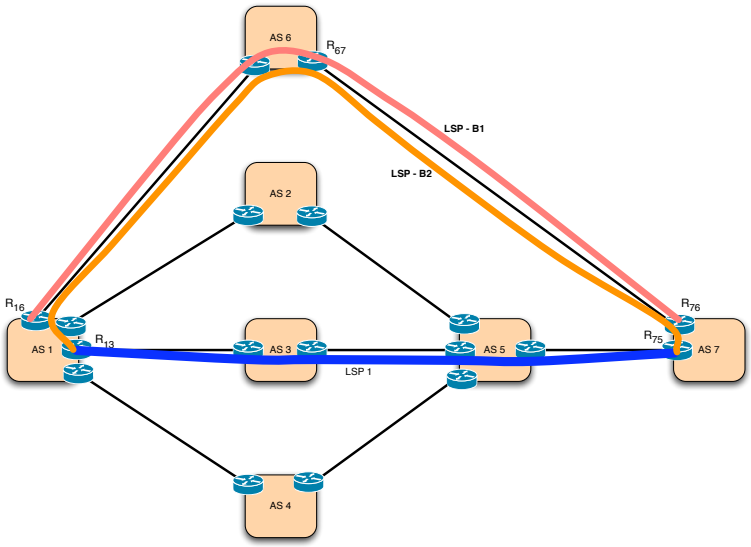


Figura 7.11: *Fast rerouting* con MPLS.

Este túnel se utilizará tanto para enviar tráfico como para poder detectar fallos en el camino y saber dónde han sido. El esquema de respaldo puede ser tanto por reencaminamiento o por protección[6], es decir, el LSP de respaldo puede haberse preestablecido o hacerlo en el momento de detección de fallo. Dependerá de la política seguida por el dominio y los acuerdos que mantenga con el resto de dominios, pero ambas posibilidades pueden llevarse a cabo, la información del AS_PATH secundario está siempre disponible.

Para crear este túnel de respaldo existen dos formas, que sea R_{16} o el propio R_{13} los que inicien la señalización de establecimiento y que, por tanto, sean el inicio del túnel de respaldo. El LSP de respaldo también se señalizará mediante RSVP-TE con los dominios *AS 6* y *AS 7* como "strict hops".

En el caso de ser R_{16} el que señalize el LSP de respaldo, LSP-B1 en la figura 7.11, haría falta modificar las rutas IGP para que los routers del dominio *AS 1* enviasen el tráfico destinado a *AS 7* hacia el router R_{16}, en lugar de hacia R_{13}. A este inconveniente en el dominio origen se une que el tráfico en el dominio destino, *AS 7*, llegaría por un router distinto al que llegaba el tráfico antes del fallo.

[6]Véase la sección 2.2 para más detalle.

El cambio de las rutas IGP puede hacerse fácilmente si tanto R_{16} como R_{13} inyectan rutas IGP destinadas a *AS 7* con mayor precedencia para las inyectadas por R_{13}. En caso de fallo, R_{13}, router que es notificado del fallo, cambiaría a una menor precedencia en dichas rutas, seleccionándose inmediatamente a R_{16} como router de salida para el tráfico a *AS 7* en todos los routers del dominio *AS 1* [89].

Que el LSP de respaldo tenga un punto de salida distinto al LSP principal en *AS 7* puede resultar una ventaja en el caso de que haya sido precisamente el router R_{75} el que haya fallado, y no el enlace entre *AS 5* y *AS 7*.

Si fuese R_{13} el que señaliza el LSP de respaldo, LSP-B2 en la figura 7.11, el proceso de reencaminamiento sería más sencillo y rápido, aunque menos óptimo. En cuanto el router R_{13} recibe la notificación del fallo puede conmutar el tráfico destinado a *AS 7* por el túnel de respaldo sin necesidad de modificar las rutas IGP y, por tanto, de manera transparente a todos los routers de *AS 1*. Esta solución no es óptima porque el tráfico dirigido a *AS 7* en lugar de ir directamente a R_{16}, debe pasar primero por R_{13}.

Para la señalización del LSP-B2 hay que utilizar un objeto ERO que debe contener como primer salto a R_{16}, después los dominios por los que debe pasar, y por último, de manera opcional, a R_{75}. El LSP de respaldo en este caso, al contrario que en el anterior, puede terminar en R_{75} porque R_{13}, como inicio del LSP principal puede conocer el LSR en el que finaliza dicho LSP e incluirlo en el objeto ERO de señalización del LSP de respaldo para que este termine en él. Esta opción sería viable para LSP-B1, cuando R_{16} es el que señaliza en LPS de respaldo, si pudiese saber cuál es el LSR final del LSP principal, R_{13} podría notificárselo. Como R_{75} es un router de borde de *AS 7* este es conocido por el resto de Sistemas Autónomos por BGP, con lo que no es información que los AS no puedan o no quieran compartir.

La elección de incluir a R_{75} o no en el objeto ERO conlleva ciertas implicaciones. Si el LSP de respaldo termina en R_{75}, durante el fallo no es necesario llevar a cabo ninguna modificación en ningún sistema. Es totalmente transparente para el resto de routers, tanto de *AS 1* como de *AS 7*, y no precisa de configuración de políticas de acceso adicionales en R_{76} que permitan recibir tráfico IP de *AS 1* desde este router. Políticas que, por otra parte, probablemente fuesen necesarias si el túnel terminase en R_{76}. Pero teniendo en cuenta que habría que permitir la construcción de LSPs de respaldo de *AS 1* que lleguen a *AS 7* por R_{76} estas políticas de acceso ya tienen que existir, por lo que en realidad no es un impedimento. La gran ventaja de terminar el LSP de respaldo en R_{76} es que este también protegería ante fallos de R_{75}.

En nuestra opinión la mejor opción es iniciar el LSP de respaldo en R_{13}, ya que involucra menos cambios dentro de *AS 1* y terminarlo en R_{76}, puesto que protege también de fallos en R_{75}.

7.3.4.3. Análisis del mecanismo presentado

En cualquiera de los casos anteriores, una vez llegada la estabilidad en las rutas BGP, se pueden reestablecer los túneles de acuerdo a las nuevas rutas BGP. Siguiendo este procedimiento la pérdida de paquetes puede llegar a ser nula, si se crea también el *reverse backup*[7], o casi nula. El tiempo de notificación y recuperación de MPLS depende fundamentalmente de la longitud entre el punto de fallo y el PSL del LSP, sobre todo si el LSP de respaldo ya está señalizado de antemano. Dentro de un dominio este tiempo puede variar entre unos pocos milisegundos y un segundo [90]. Si hay varios dominios involucrados el tiempo de recuperación a lo sumo sería de algún segundo. Entonces, con un tiempo de no más de uno o dos segundos el tráfico interdominio está restaurado en cualquier situación. Merece la pena resaltar este tiempo de reacción comparado con las varias decenas de minutos que pueden llegar a tardar en estabilizarse las rutas BGP [6].

La clave de esta recuperación rápida está en que gracias al LSP que hay formado es posible conocer que existe un fallo y a qué tráfico afecta. El router de entrada al túnel es informado en poco tiempo del fallo, pudiendo reaccionar este rápidamente.

El túnel de respaldo puede estar preestablecido o no. Las ventajas y desventajas de la protección vs. el reencaminamiento se discutieron en el apartado 2.2 y son válidas en este caso. Sólo cabe comentar si es posible realizar un sistema basado en reencaminamiento (cálculo y señalización bajo demanda).

Debido a que el mensaje de NOTIFY llega al router de entrada al túnel, R_{13}, antes de que se inicie el reajuste de rutas BGP, este todavía dispone de las rutas secundarias válidas para poder señalizar en ese momento los LSPs de respaldo. Es posible, por tanto, utilizar tanto un esquema de protección como de reencaminamiento, quedando la elección al operador de red conforme a los requisitos de la misma.

Una de las principales ventajas de disponer de caminos disjuntos a nivel de Sistema Autónomo es que no es necesario compartir ningún tipo de información adicional entre los diferentes dominios. Con la información que comúnmente se intercambian los ASes entre sí, *AS Number* e identificadores de los routers de borde, es suficiente, como hemos visto, para poder establecer los túneles MPLS principales y de respaldo. No es necesario proveer ningún mecanismo de ocultación de información en un AS que utilice el mecanismo descrito.

En este nuevo mecanismo también hay lugar para las políticas de control y admisión. En el ejemplo de la figura 7.10 *AS 7* debe permitir terminar túneles de respaldo en su dominio iniciados por *AS 1*, además *AS 6* debe permitir ser tránsito entre *AS 1* y *AS 7* en caso de fallo.

Todos los mecanismos propuestos hasta la fecha, especialmente PCE, presupo-

[7] Véase la sección 2.3.

nen que el mecanismo utilizado sólo será utilizado en un conjunto reducido de ASes, no en todo Internet. Normalmente se justifica indicando que por problemas de escalabilidad no es viable manejar la información requerida por dicha solución de todos los ASes.

En el caso del esquema propuesto en esta Tesis Doctoral, el tiempo de convergencia y tamaño de las tablas de rutas para las rutas principales (rutas "clásicas" de BGP) no se ve alterado. Esto es debido a que el proceso de selección y envío de las rutas principales no se ha visto modificado por el proceso de selección y envío de rutas secundarias[8].

Respecto a la convergencia y tamaño de las rutas secundarias depende del número de ASes involucrados. En el caso peor, con todo Internet, el tamaño de las tablas de encaminamiento secundarias y el tiempo de convergencia son similares a los de los principales. Un AS, para un determinado destino, tendrá como mucho una entrada por cada AS con el que haga *peering* en las RIB de rutas secundarias, al igual que en las RIB de rutas BGP convencionales. Manejando la misma cantidad de rutas y teniendo el mismo número de *peers* el tiempo de convergencia para obtener las rutas secundarias será similar al de obtener las rutas principales. Eso sí, hasta que no se tienen rutas principales seleccionadas en los routers no se pueden empezar a obtener las secundarias, por tanto, el tiempo de convergencia es similar pero todo el proceso se pospone hasta que las RIB tienen las rutas BGP principales.

En el caso peor, la implantación en toda la internet del mecanismo, la escalabilidad de la propuesta está supeditada a la de BGP, incrementando el uso de memoria, red y tiempo, como mucho, al doble de lo consumido por BGP "clásico".

7.3.4.4. Acuerdos entre ASes

Ocurre habitualmente que los diferentes Sistemas Autónomos tienen contratos particulares con otros Sistemas Autónomos. Estos acuerdos definen qué tipo de tráfico pueden intercambiarse entre ellos y en qué cantidad. Es difícil creer que cualquier AS pueda realizar túneles MPLS (o de cualquier otro tipo) que finalicen en cualquier otro AS o atraviesen cualquier AS. El modelo de colaboración será parecido al actual, explícito. Modelo en el cuál sólo los ASes con un acuerdo podrán finalizar túneles interdominio en los routers de otro dominio. Siguiendo este modelo, lo mismo aplica para los túneles de respaldo, sólo aquellos ASes con acuerdos podrán enviar tráfico de respaldo entre ellos. Se formaran pequeñas "islas" de ASes con acuerdos, en las que algunos de los ASes pueden formar parte de varias "islas", haciendo de tránsito en los respaldos.

En la figura 7.12 puede verse un ejemplo de dos "ìslas" diferentes de ASes que

[8] Sí se ve alterado por un aumento del tráfico de control.

tienen acuerdos entre sí. Los *AS 4* y *AS 5* forman parte de las dos "ìslas". También aparecen en la figura las rutas BGP principal y secundaria de cada AS cuando el destino es *AS 8*, y en el caso del *AS 1* aparecen este par de rutas para todos los AS de la figura. Por supuesto, los AS que no pertenecen a la *"isla" B* no tienen rutas BGP secundarias para estos AS (en particular a *AS 8*).

El aumento de las RIB secundarias y la convergencia para obtener las rutas secundarias es la correspondiente a 5 dominios en cada "isla", tamaño y tiempo despreciable frente al de manejar las inmensas tablas BGP de cualquier dominio.

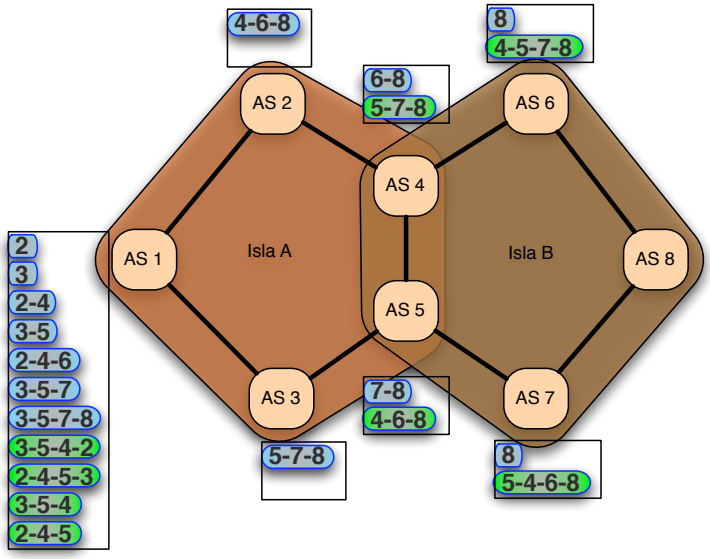


Figura 7.12: Grupos de Sistemas autónomos con acuerdos distintos.

Se va a mostrar, con un ejemplo basado en la figura 7.12, que incluso en el caso peor, en el que todo un AS deja de funcionar la funcionalidad de respaldo que no son de tu isla no se pierde. Tal y como está en la RIB de *AS 1*, éste envía el tráfico a *AS 7* y a *AS 8* a través de *AS 5*. Debido a que entre *AS 1* y *AS 7* o *AS 8* no hay acuerdo alguno *AS 1* encaminará el tráfico hacia estos destinos por el LSP abierto con *AS 5*.

Si sucede un fallo en el enlace entre *AS 5* y *AS 7* será *AS 5* quien encamine todo el tráfico destinado a *AS 7* y *AS 8* por sus túneles de respaldo a dichos ASes, incluido el que le ha enviado *AS 1* por medio del LSP que les conecta.

Si el fallo se produjese en todo el *AS 5*, entonces, *AS 1* utilizaría el LSP de respaldo que pasa por *AS 4* hasta *AS 5*[9]. Si el fallo es de todo *AS 5* entonces *AS 1* recibirá otra notificación de fallo por el LSP de respaldo. Esto le puede hacer sospechar que el fallo no es un enlace si no todo el AS. Con el tráfico a *AS 5* no hay nada que hacer,

[9] *AS 1* no puede saber si el fallo es de todo *AS 5* o sólo el acceso desde *AS 3*.

pero todo el resto de tráfico que simplemente pasa por *AS 5* puede reencaminarse hasta el AS anterior del túnel de respaldo entre *AS 1* y *AS 5*, esto es, hasta *AS 4*. Para ello se puede utilizar el LSP que lleva hasta él [10]. Una vez el tráfico en *AS 4* este utilizará sus LSPs principales o de respaldo para alcanzar los diferentes destinos.

Este problema sólo ocurre cuando un AS que forma parte de dos "islas" (*AS 5*) falla y hay que reenviar el tráfico no destinado a ese AS por uno de los otros AS que conecta a otra "isla" (*AS 4*). Utilizar el AS anterior en el LSP de respaldo o principal no siempre tiene por qué funcionar. La mejor opción sería que los ASes supiesen cuáles de los ASes con los que tiene acuerdos pertenecen a más de una "isla".

En todos los casos descritos de utilización de la ruta de respaldo se ha indicado su utilización en cuanto el fallo es detectado. Después del mismo, la utilización del LSP de respaldo puede prolongarse hasta que el fallo es arreglado, hasta que las nuevas rutas BGP son estables y se pueden volver a utilizar o en cualquier momento intermedio en el que se quiera optimizar la utilización de la red. La decisión será del operador de la misma.

7.4. Conclusiones

En este capítulo se ha presentado una modificación al mecanismo de selección de rutas de BGP. Mediante esta modificación los diferentes Sistemas Autónomos de la red pueden obtener, además del AS_PATH convencional de BGP para cada destino, un nuevo AS_PATH para cada destino, denominado AS_PATH_DISJ, que es disjunto en ASes al AS_PATH seleccionado por el AS para dicho destino.

Se ha explicado como utilizando BGP convencional no es posible obtener dos secuencias de AS disjuntas para llegar a cada destino, y cómo el mecanismo propuesto sí. Para ello se han mostrado diferentes ejemplos de funcionamiento del mecanismo, explicando paso a paso el intercambio de mensajes entre los distintos Sistemas Autónomos y el contenido de las RIBs de dichos ASes.

Se ha demostrado analíticamente que si la red de ASes es 2-conectada todos los dominios obtienen dos rutas, AS_PATH y AS_PATH_DISJ, disjuntas entre sí al resto de dominios.

Se ha presentado un ejemplo de uso de dicho mecanismo, y cómo utilizando MPLS y las rutas computadas por el mecanismo propuesto es posible implementar un mecanismo de reencaminamiento rápido interdominio.

Durante el análisis se ha señalado que el consumo de recursos, tanto en tiempo como en memoria, necesarios para el cómputo de las rutas disjuntas es similar al

[10] Entre el principal y el de respaldo el que todavía este conectado, ya que uno de ellos puede haber dejado de funcionar si pasa por *AS 5*, como, por ejemplo, en este caso el LSP de respaldo.

necesario para computar las rutas BGP convencionales. No obstante, buscando una solución realista se ha analizado el caso en el que los dominios no aceptan LSPs extremo a extremo desde cualquier otro dominio, sólo de aquellos con los que tiene un acuerdo, con los que forma una "isla". En este caso también es posible tener mecanismos rápidos de reencaminamiento entre ASes aunque no pertenezcan a la misma "isla". Para que esto sea posible en todos los casos, independientemente de si falla un enlace o todo un AS, es necesario que haya al menos dos ASes de una "isla" que formen parte de otra "isla".

El trabajo presentado en este capítulo cubre una carencia conocida de BGP, debido a la agregación de rutas, que facilita la ingeniería de tráfico multidominio y, en particular, permite la utilización de mecanismos de recuperación rápida frente a fallos en recursos, routers o enlaces, interdominio.

El mecanismo de selección de rutas presentado es coherente y respeta las políticas y acuerdos entre sistemas autónomos, del mismo modo en que BGP lo es.

Hasta donde llega nuestro conocimiento, el trabajo aquí presentado, incluidos los teoremas y demostraciones, son totalmente novedosos. Además es la primera propuesta escalable y distribuida que permite obtener AS_PATHs disjuntos entre los diferentes Sistemas Autónomos sin interferir en el funcionamiento normal de BGP, incluidas las políticas internas y de interdominio de los Sistemas Autónomos.

Este mecanismo permite además recuperarse ante la caída de todo un Sistema autónomo, y no sólo de un enlace interdominio, como ocurre con las estrategias planteadas en la literatura (sección 2.4).

Capítulo 8

Consideraciones sobre los p-ciclos multidominio

En este capítulo se van a realizar una serie de consideraciones acerca de la posible utilización de los p-ciclos para la protección interdominio. Se verá también qué es necesario para poder llevarlo a cabo y cómo podría ser el mecanismo de reacción en caso de que hubiese un fallo en un enlace interdominio.

8.1. P-ciclos interdominio

Realizar un p-ciclo multidominio entre un conjunto de dominios consiste en seleccionar qué enlaces interdominio se utilizarán para obtener el ciclo de protección. Esto suponiendo que se conozca cuáles son los dominios que van a formar parte de el p-ciclo, puesto que es posible proteger varios dominios con más de un p-ciclo [91]. Utilizar más de un p-ciclo para proteger puede hacerse por diversos motivos, los principales son disminuir el tiempo de reacción y minimizar la utilización de recursos [92]. En todo caso, una vez decididos qué dominios van a protegerse con un p-ciclo es necesario encontrar un p-ciclo que los recorra a todos. Es decir, es necesario encontrar un AS_PATH que recorra todos los dominios. Para ello es necesario poder encontrar dos AS_PATHs disjuntos que vayan de un dominio a otro, puesto que estos formarán un ciclo. Como ya se ha comentado en esta Tesis Doctoral hay diversas razones por las que con el protocolo de encaminamiento actual utilizado en Internet, BGP, esto no es posible (sección 2.5) [82] [79].

Aún así existen tres posibles formas de obtener estos ciclos, modificar BGP, utilizar PCE o utilizar una configuración manual.

1. El principal problema de BGP es que es imposible conocer, fundamentalmen-

te por razones de escalabilidad, el grafo completo de dominios de la red, cómo están interconectados entre sí. El número de caminos distintos que ofrece BGP entre dos dominios dados es uno. Con esta limitación no es posible realizar ingeniería de tráfico a nivel de ASes. Pequeñas modificaciones, como la propuesta en esta Tesis Doctoral en el capítulo 7, en BGP pueden permitir tener un mayor conocimiento de esta red de dominios permitiendo realizar algo de ingeniería de tráfico. En particular, con la propuesta de esta Tesis Doctoral, obtener AS_PATHs disjuntos entre ASes que permitirían construir un p-ciclo.

Con este tipo de modificaciones no es posible tener conocimiento de todo el grafo de modo que permita realizar verdadera ingeniería de tráfico, es decir, con las modificaciones propuestas, por ejemplo, no es posible decidir con total libertad en qué orden o cuáles son los dominios que formarán el p-ciclo, y en general será así pues si no estaríamos ante una solución no escalable. Por tanto, este tipo de solución sólo sería válida si se quisiese que el p-ciclo atravesase por algunos dominios sin importar si también protege a otros dominios o en qué orden lo hace, no si se quisiese definir exactamente cuáles son los ASes involucrados de antemano y en qué orden se quieren proteger [91].

2. Como también se ha comentado en esta Tesis Doctoral (sección 2.4) en el IETF se está definiendo el PCE. Se trata de una arquitectura diseñada para computar caminos, tanto intradominio como interdominio. Para la computación interdominio, el propio grupo del IETF indica que, está limitada a unos pocos dominios. Estos dominios deben compartir alguna información entre ellos, es por esto, razones de seguridad y escalabilidad, que sólo unos pocos dominios pueden estar involucrados entre sí [16].

   Si se quiere proteger con un p-ciclo un conjunto de ASes es posible formar un "grupo PCE". El grafo completo de unión entre estos dominios puede ser compartido y así obtener todos los posibles p-ciclos que los involucren y elegir el más adecuado en cada momento.

   Esta es, sin duda, la opción óptima para un grupo pequeño y predefinido de ASes. No hay que olvidar que están presentes los problemas ya comentados para PCE, fundamentalmente, es necesario extender la arquitectura de los dominios involucrados, además de establecer lazos de confianza entre ellos. Otro punto en contra a tener en cuenta es que PCE todavía no está definido totalmente, ni la arquitectura, ni los protocolos, ni los procedimientos, con lo que su utilización queda para un incierto futuro.

3. Por último, y actualmente la más viable de todas, es la configuración manual de los p-ciclos. Los cambios topológicos de los enlaces de interconectividad entre un conjunto reducido de dominios no es frecuente. Los enlaces pueden fallar y recuperarse, pero un cambio permanente, un enlace añadido o uno eliminado, ocurre raramente. En el caso en el que los p-ciclos se utilizan como

mecanismo de protección no es necesario recalcular el p-ciclo durante el fallo de un enlace. Por este motivo los p-ciclos utilizados para protección interdominio entre un conjunto conocido de antemano de dominios pueden calcularse y configurarse manualmente.

Si el conjunto de dominios no es conocido de antemano o se necesita una solución automática (no manual) entonces hay que recurrir a una de las propuestas anteriores.

Tan pronto como el AS_PATH de dominios para el p-ciclo es computado hay que decidir por dónde se establecerá, a nivel interdominio, exactamente el p-ciclo, esto es, cuáles de los enlaces interdominio se utilizarán. Una vez que se tiene toda esta información es posible establecer el p-ciclo.

Hasta ahora sólo se ha tenido en cuenta el p-ciclo en el interdominio, qué dominios recorre y qué enlaces utiliza, pero no cómo se unen los enlaces interdominio internamente entre sí , esto es, el enlace del p-ciclo que accede a un dominio con el enlace del p-ciclo que sale del dominio. Esta unión puede realizarse de múltiples maneras, por ejemplo, utilizando caminos interiores directos o p-ciclos intradominio [91].

8.2. Ejemplos de fallo de enlaces

En esta sección se van a exponer dos ejemplos de cómo se podrían utilizar los p-ciclos en caso de fallo en un enlace interdominio. La figura 8.1 muestra el p-ciclo que se ha configurado para los cinco dominios de la figura. El p-ciclo configurado utiliza los enlaces interdominio D1-D2-D3-D5-D4-D1, protegiendo estos y además protegiendo el enlace interdominio D2-D4, que no forma parte del p-ciclo pero que queda protegido por él[1].

En el primer ejemplo se verá la señalización necesaria cuando falla un enlace que forma parte del p-ciclo. En el segundo veremos la señalización cuando falla un enlace que no pertenece al p-ciclo.

Fallo de un enlace interdominio que forma parte del p-ciclo Si falla un enlace interdominio del p-ciclo, por ejemplo el enlace D1-D4 de la figura 8.2, algunos LSPs multidominio pueden verse afectados, como por ejemplo el LSP 1 en la figura. Cuando el router R_{14} detecta el fallo este envía un mensaje TE de tipo *Notify* por el p-ciclo para informar de que se ha detectado un fallo y que el p-ciclo está en

[1] En la literatura a este tipo de enlaces se les denomina *"straddling links"*.

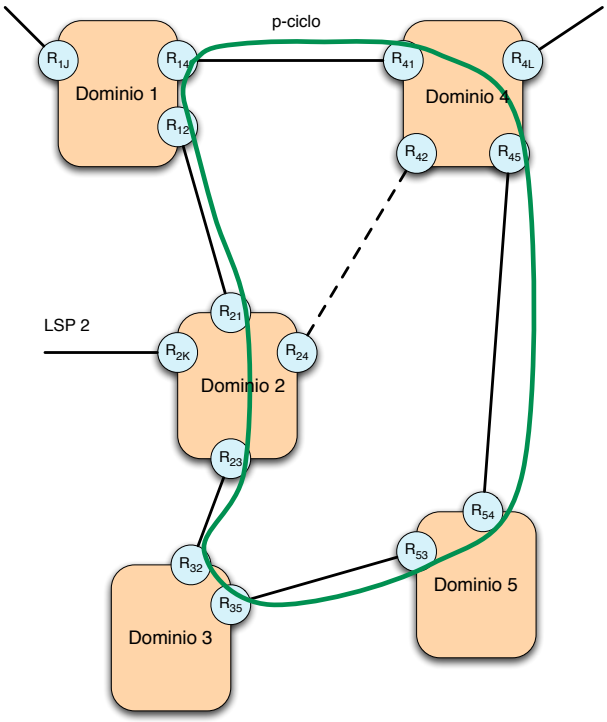


Figura 8.1: P-ciclo multidominio protegiendo 5 dominios.

uso. Este mensaje alerta al router R_{41} de que debe extraer los paquetes del p-ciclo y volver a inyectarlos en el LSP 1. Inmediatamente los routers R_{14} y R_{41} reencaminan los paquetes del LSP 1 al p-ciclo y de este al LSP 1 respectivamente. Por medio de este mecanismo ningún otro router del LSP es consciente del fallo.

El mensaje de *Notify* además de alertar a estos dos routers de la nueva situación alerta al resto de routers de que el p-ciclo ya está en uso y que no puede ser utilizado por otros routers. Cabe recordar que los p-ciclos sólo protegen de un fallo simultáneamente. Existen algunas optimizaciones que permiten que en algunas circunstancias se pueda proteger más de un fallo [93].

Fallo en un enlace interdominio que no forma parte del p-ciclo Si el que falla es un enlace interdominio que no pertenece al p-ciclo, como el enlace D2-D4 de la figura 8.3, la señalización requerida es un poco más compleja. Cuando el router R_{24} detecta el fallo, los LSPs afectados por el fallo, como LSP 2 en la figura, deben reencaminar los paquetes hasta el p-ciclo. Para poder realizar esto, el router interdominio, que no pertenece al p-ciclo, debe conocer cuál es el router más cercano del dominio que sí pertenece al p-ciclo. Entonces, el router R_{24} envía a este router, R_{23} en la figura, un mensaje TE *Notify* para indicarle que le va a reenviar el tráfico del

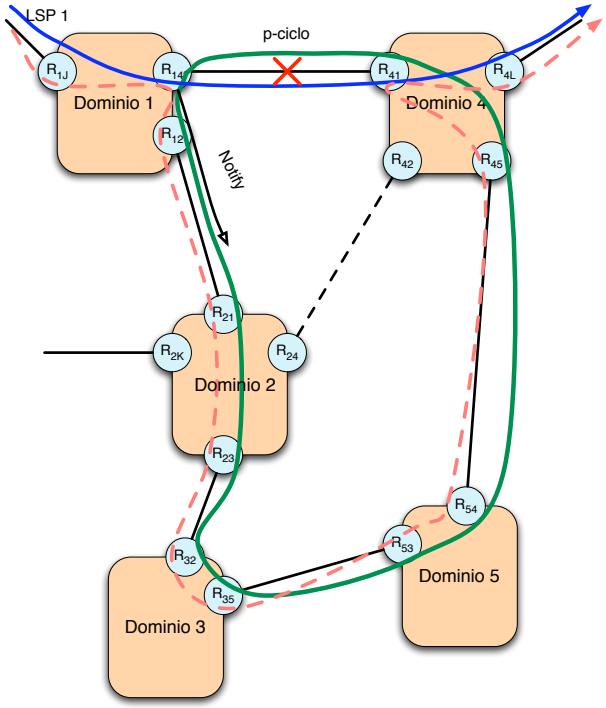


Figura 8.2: Fallo de un enlace del p-ciclo.

LSP 2 va para que lo conmute por el p-ciclo y que el siguiente salto del LSP 2 es R_{42} del Dominio 4. R_{24} conmuta el LSP 2 hacia R_{23} por medio de un LSP intradominio.

El router R_{23} recibe el mensaje de *Notify* y los paquetes del LSP 2 desde R_{24}. A su vez envía un mensaje TE *Notify* por el p-ciclo para informar de un fallo en un enlace de tipo "straddling", que el p-ciclo está en uso y que el router del Dominio 4 debe extraer el tráfico del p-ciclo y reenviarlo al router R_{42}. El router R_{45} envía el tráfico a R_{42} por medio de un LSP intradominio y este conmuta el tráfico al LSP 2, restaurando el camino normal. Todo esto ocurre sin que el resto de routers del LSP 2 se percaten del fallo.

Cabe hacer notar que la dificultad aquí reside en que los routers que no forman parte del p-ciclo deben tener conocimiento de la existencia de este y cuál es su router de conexión con el p-ciclo. Este conocimiento sólo deben tenerlo los router frontera de los dominios que además quieran proteger sus enlaces interdominio. Los router intradominio que no están involucrados en este mecanismo de protección interdominio y aquellos routers que no se quiera proteger (y que no formen parte del p-ciclo) tampoco tienen por qué tener conocimiento de la existencia del p-ciclo, por lo que no parece que sea un gran inconveniente configurar aquellos nodos en los que se quiera proteger sus enlaces interdominio.

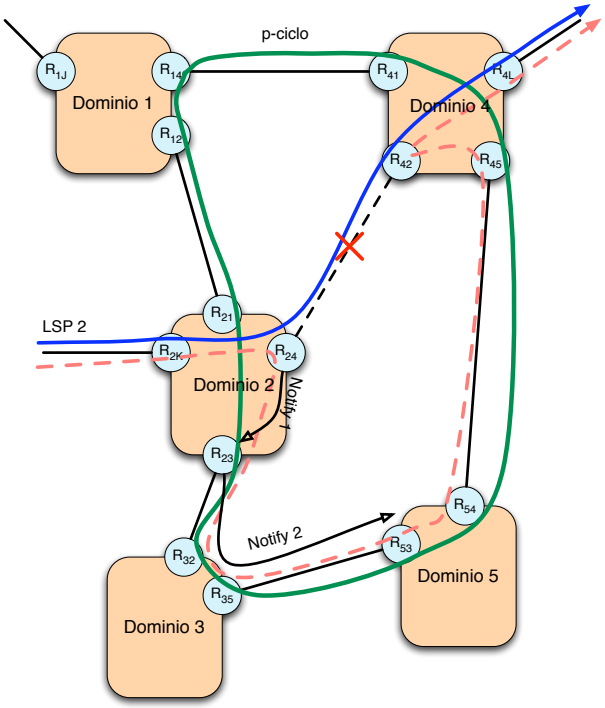


Figura 8.3: Fallo de un enlace que no es parte del p-ciclo.

En este último caso de fallo además es posible llevar a cabo algún tipo de optimización, por ejemplo, en el Dominio 2 el tráfico del LSP 2 puede conmutarse desde R_{2K} a R_{23} directamente en lugar de desde R_{24} a R_{23}. En el Dominio 4 podría realizarse el mismo tipo de optimización.

8.3. Conclusiones

Los p-ciclos pueden resultar una herramienta interesante para la protección multidominio, pero todavía tienen algunas inconvenientes que salvar.

Son interesantes porque la gestión de los mismos es sencilla, y los routers involucrados son los imprescindibles para brindar la protección.

Por contra, de momento, la única opción viable para su cómputo y establecimiento es la manual. Además utilizando p-ciclos sólo es posible proteger los enlaces interdominio, puesto que para proteger también los nodos, en general, utilizando p-ciclos son necesarios los llamados p-ciclos “rodeando un nodo” cuya construcción en un entorno multidominio sería muy compleja [93].

Parte III

Conclusiones finales y trabajo futuro

Capítulo 9

Conclusiones

En este capítulo vamos a resumir las principales conclusiones que se desprenden de la presente Tesis Doctoral .

En la presente Tesis Doctoral se han propuesto diversas soluciones al problema de proteger LSPs MPLS interdominio. Para ello, se analizaron previamente los esquemas, tanto intradominio como interdominio, más importantes propuestos hasta la fecha.

Las soluciones propuestas pueden dividirse en dos tipos, búsqueda de caminos disjuntos por secuencias de ASes disjuntos y búsqueda de caminos disjuntos por la misma secuencia de ASes.

Para la primera de ellas, es necesario modificar el protocolo de encaminamiento interdominio actual, BGP-4. En esta Tesis Doctoral se ha presentado una modificación al mecanismo de decisión de rutas de BGP. Mediante esta modificación los diferentes Sistemas Autónomos de la red pueden obtener, además de un AS_PATH convencional BGP para cada destino, un nuevo AS_PATH para cada destino, denominado AS_PATH_DISJ, que es disjunto en ASes al AS_PATH computado por el AS para dicho destino.

Se ha mostrado cómo utilizando BGP convencional no es posible obtener dos secuencias de AS disjuntas para llegar a cada destino, y cómo el mecanismo propuesto sí, sin alterar los tiempos de convergencia de la ruta principal.

Se ha demostrado analíticamente que si la red de ASes es 2-conectada todos los dominios obtienen dos secuencias de ASes a cada uno de los dominios denominadas AS_PATH y AS_PATH_DISJ que son disjuntas entre sí.

Se ha presentado cómo es posible, utilizando MPLS y las rutas computadas por el mecanismo propuesto, implementar un mecanismo de reencaminamiento rápido multidominio.

Se ha analizado el consumo de recursos, tanto en tiempo como en memoria, necesarios para el cómputo de las rutas disjuntas y se ha visto que es similar al necesario para computar las rutas BGP convencionales.

También se ha analizado el caso en el que los dominios no aceptan LSPs extremo a extremo desde cualquier otro dominio, sólo de aquellos con los que tiene un acuerdo, con los que forma, lo que se ha llamado, una "isla". En este caso también es posible tener mecanismos rápidos de reencaminamiento entre ASes aunque no pertenezcan a la misma "isla". Se ha mostrado que para que esto sea posible en todos los casos, independientemente de si falla un enlace o todo un AS, es necesario que haya al menos dos AS de una "isla" que formen parte de otras "islas".

El mecanismo BGP aquí presentado es totalmente novedoso. Es la primera propuesta escalable y distribuida que permite obtener AS_PATHs disjuntos entre los diferentes Sistemas Autónomos.

Este mecanismo permite además recuperarse ante la caída de todo un Sistema autónomo, y no sólo de un enlace interdominio, como ocurre con las estrategias planteadas en la literatura (sección 2.4) o en el resto de propuestas de esta Tesis Doctoral.

Dentro de la segunda opción se han propuesto dos esquemas. Un primer esquema que computa caminos disjuntos extremo a extremo, sin caer en trampas topológicas y teniendo en cuenta la posible división de un dominio en distintas áreas IGP. Y un segundo esquema que tiene en cuenta los caminos de respaldo ya establecidos en el interior de los dominios y los utiliza para proteger las zonas intradominio, preocupándose especialmente de proteger la zona interdominio.

Para el primer esquema ha sido necesario presentar y demostrar previamente un algoritmo de cómputo de caminos disjuntos con características especiales.

El algoritmo presentado computa dos caminos disjuntos dentro de un dominio, desde dos nodos de entrada a dos nodos de salida. Para poder realizar estos cálculos el algoritmo se basa en los caminos computados de manera independiente por cada uno de los nodos de entrada al dominio. Esto permite poder computar caminos disjuntos dentro del dominio sin necesidad de compartir toda la información topológica de la red.

El algoritmo de cómputo propuesto respeta al máximo las decisiones locales de los nodos de entrada, intentado ajustarse al máximo a los caminos propuestos por los nodos de entrada.

Se ha visto que utilizando esta metodología es posible computar un par de caminos disjuntos siempre que estos existan, es decir, si hay una solución el algoritmo es capaz de computarla sin caer en ninguna trampa y, por tanto, sin necesidad de utilizar mecanismos de *backtracking*.

Utilizando el algoritmo propuesto se ha presentado un mecanismo para el cómputo de caminos disjuntos extremo a extremo que crucen por la misma secuencia de Sistemas Autónomos. Se ha explicado también el mecanismo intradominio que hay que llevar a cabo en un dominio dividido en diversas áreas IGP para obtener y distribuir la información de encaminamiento necesaria. Una vez que este mecanismo interior está en marcha, los LSRs de entrada a los diferentes dominios disponen de la información requerida en el procedimiento de cómputo del mecanismo multidominio.

Este mecanismo multidominio puede establecer los caminos principal y de respaldo con un solo intercambio de mensajes de señalización (PATH + RESV), también con dos. En este caso, se ha visto, que los caminos computados en cada uno de los distintos dominios deben ser enviados en los mensajes de PATH y que, por tanto, es necesario compartir información topológica entre los diferentes dominios, por lo que realmente no es aconsejable su uso.

También se ha visto que con el coste de utilizar dos intercambios de mensajes de señalización, el primero para obtener los nodos interdominio a utilizar y el segundo para establecer los caminos principal y de respaldo, es posible computar y establecer los caminos disjuntos multidominio extremo a extremo sin necesidad de compartir información topológica entre dominios, por lo que se considera más adecuado utilizar este mecanismo de dos intercambios de mensajes.

En cualquiera de ambos casos, por el hecho de utilizar el algoritmo es posible afirmar que si existe solución al problema este mecanismo es capaz de encontrarla en primera iteración, sin caer en trampas topológicas.

Se puede decir, entonces, que, hasta donde llega nuestro conocimiento, este mecanismo es el primero propuesto para el cómputo y establecimiento de LSPs disjuntos multidominio que es distribuido, evita las topologías trampa, permite la ocultación de las topologías entre dominios, tiene en cuenta la división en áreas IGP de los dominios y es escalable.

El segundo mecanismo propuesto es un nuevo esquema de respaldo interdominio en el que se tiene en cuenta que los dominios realizan recuperación “local”, esto es, protegen sus propios recursos de manera interna.

Se han mostrado los posibles esquemas de respaldo que se pueden utilizar en el intradominio, y cómo independientemente del utilizado, estos son capaces de proteger frente al fallo en cualquier recurso del dominio, excepto en los LSRs frontera. Tampoco es posible proteger desde el interior de un dominio los enlaces interdominio. Por tanto, estos son los únicos recursos, nodos y enlaces interdominio, que se deben proteger.

Debido a que sólo hay que proteger los recursos interdominio se relajan las restricciones que debe cumplir el LSP de respaldo respecto al LSP principal. Sólo es

necesario que sean disjuntos en el interdominio. Además, no es necesario compartir información privada interior entre los dominios.

El LSP principal siempre es el óptimo, puesto que al no tener restricciones dentro de los dominios no hay problema de caer en una trampa topológica y el LSP de respaldo puede establecerse de manera independiente al principal dentro de los dominios.

Se ha realizado un análisis cualitativo y cuantitativo en el consumo de recursos empleando este esquema, resultando en un ahorro de recursos si los LSPs de respaldo "*interior*" y extremo a extremo comparten recursos.

Si en una red hay implantado MPLS en los routers es más que probable que una de las aplicaciones que se utilice sea la protección de enlaces dentro del dominio, y con esta la de LSPs. Suponiendo pues que es el propio dominio quien protege los LSPs (intradominio o multidominio) ante fallos dentro de la red, los LSPs de respaldo multidominio no deberían preocuparse de proteger estos recursos. Sólo deberían preocuparse de aquellos que los dominios por sí mismos no pueden proteger, y requieren de colaboración con otros dominios para ello. Estos recursos son los del interdominio. Por lo que este esquema de respaldo propuesto en la presente Tesis Doctoral puede ser válido en un gran número de escenarios.

Ante el fallo de un recurso en el interior de un dominio se utilizan los mecanismos de recuperación intradominio, en el caso de un fallo en un recurso interdominio se utiliza el LSP de respaldo multidominio. La ventaja de este esquema es que un fallo dentro de un dominio no trasciende al resto de dominios.

En los esquemas de respaldo multidominio "tradicionales" un fallo en cualquier nodo o enlace del LSP principal provoca que se cambie al LSP de respaldo. Esto requiere que cualquier fallo interno de un dominio se notifique al LSR origen y, por tanto, que la notificación de fallo se transmita por otros dominios.

Para terminar, se puede decir que esta Tesis Doctoral propone tres esquemas distribuidos de respaldo para comunicaciones multidominio. Entre los tres esquemas se abarcan las diferentes posibilidades de un modo totalmente novedoso, distribución de rutas disjuntas por BGP, computo y señalización de caminos disjuntos extremo a extremo sin caídas en trampas topológicas y sin necesidad de compartir información topológica entre dominios y computo y señalización del camino principal y de respaldo que protege la zona interdominio y colabora con la protección "local" intradominio para aumentar la eficiencia en la reserva de recursos.

Capítulo 10

Trabajo futuro

La continuación del trabajo desarrollado en esta Tesis Doctoral pasa por publicar sus resultados en revistas de alto impacto así como enviar algún Internet Draft que explique los resultados de esta Tesis Doctoral. En particular la propuesta de modificación de BGP-4 de modo que se permita la obtención de AS_PATHs disjuntos entre distintos Sistemas Autónomos.

Junto a la publicación de los resultados sigue la continuación de los trabajos de investigación para ampliar el marco de los escenarios en los que se pueden aplicar los mecanismos aquí propuestos. Viendo si modificar el escenario implica modificar de algún modo los mecanismos y por tanto necesitar una generalización de los mismos.

Un trabajo muy interesante y muy novedoso sería realizar un análisis de cómo se comportarían los mecanismos propuestos ante peticiones de caminos disjuntos con restricciones de QoS. Muy probablemente fuese necesario modificar los mecanismos o restringir el escenario de aplicación a uno más sencillo.

Estudios preliminares nos han permitido ver que es posible aplicar el algoritmo de cómputo de caminos disjuntos en redes cuyo protocolo de encaminamiento esté basado en vector distancias. Un estudio más profundo en este tema es necesario para buscar las limitaciones tiempos de búsqueda ahorro frente a una búsqueda por inundación, etc.

Capítulo 11

Lista de contribuciones

Las principales contribuciones de esta Tesis Doctoral son:

Algoritmo APD Se ha propuesto un nuevo algoritmo de cómputo distribuido de caminos disjuntos que evita topologías trampa mediante el cual es posible obtener caminos disjuntos entre dos pares de nodos (dos de entrada y dos de salida). Cada uno de los nodos de entrada calcula caminos disjuntos a los nodos de salida (pueden ser más de dos) desde sí mismos. Con los caminos calculados de manera independiente por cada uno de los nodos de entrada es posible seleccionar cuáles son los nodos de salida más adecuados así como cómo obtener los caminos disjuntos a ellos desde los nodos de entrada seleccionados. Se han mostrado ejemplos de utilización y se ha demostrado su validez matemáticamente así como su capacidad para encontrar siempre una solución cuando esta existe.

Mecanismo en escenarios multiárea utilizando el APD Partiendo del funcionamiento del APD se ha diseñado un mecanismo de intercambio de información entre routers de borde de diferentes áreas de un mismo dominio de modo que los routers frontera del dominio sean capaces de computar dos caminos disjuntos desde un par de routers frontera de entrada al dominio a otro par de routers frontera de salida del dominio pasando por las diferentes áreas de un sólo paso, sin necesidad de iniciar una petición de computación que englobe a todo el dominio.

Mecanismo en escenarios multiAS utilizando el APD Utilizando el APD no es necesario conocer toda la topología de red para poder computar caminos disjuntos multidominio. Partiendo de esta idea se ha diseñado un mecanismo de intercambio de información entre diferentes dominios para poder realizar este cómputo de un modo sencillo. Este mecanismo permite a su vez mantener oculta la información interior de cada dominio al resto de los mismos.

Mecanismo eficiente de protección multidominio reutilizando recursos de los dominios Si se quiere realizar protección interdominio es factible pensar que en el interior de los dominios también se realiza protección de enlaces, segmentos o caminos completos. Si esto es así, realmente no es necesario que todo el camino interdominio extremo a extremo sea protegido con otro camino de respaldo completamente disjunto. Los recursos interiores de un dominio quedan protegidos por ellos. En esta Tesis Doctoral se propone un mecanismo mediante el cuál solo se protege la zona interdominio delegando la protección interior a cada uno de los dominios. De este modo el uso de recursos en el interior de los dominios es más eficiente y la creación del camino de respaldo multidominio mucho más sencilla.

Propuesta de modificación de BGP que permite obtener AS_PATHs disjuntos entre cada par de Sistemas Autónomos Siempre que se habla de caminos disjuntos multidominio para protección se piensa en caminos disjuntos que transcurren por la misma secuencia de ASes. Esto es debido a que en el estado del arte es difícil encontrar alguna propuesta que difiera de esta visión del problema. Y es difícil encontrarla porque la información de que dispone un dominio impide poder elegir la secuencia de ASes para alcanzar un destino. Esta limitación viene impuesta por el protocolo de encaminamiento interdominio, BGP. En esta Tesis Doctoral se ha propuesto una modificación al protocolo BGP que permite obtener a los dominios dos secuencias de ASes disjuntas para alcanzar a cualquier otro dominio. Obteniendo un camino por cada una de estas secuencias se obtienen dos caminos intrínsecamente disjuntos. Todo esto con una complejidad adicional, consumo de recursos, etc. que no supera en ningún caso el doble de lo consumido por BGP actualmente. También se propone cómo puede ser utilizado sólo entre un conjunto reducido de dominios y no entre todos los dominios de la Red, con lo que el consumo de recursos extra es mínimo y más acorde a los objetivos de todos los mecanismos propuestos, en los que siempre se indica que sólo es posible entre ASes que hayan llegado a un acuerdo entre ellos.

Parte IV

Apéndices

Apéndice A

Redes 2-conectadas

Definición A.1. *Una **red** es un conjunto de nodos y enlaces.*
Red $=< \mathscr{N}, \mathscr{L} >$
donde, $\mathscr{N}$ *es el conjunto de nodos de la red con capacidad de reconfiguración y* $\mathscr{L}$ *es el conjunto de enlaces de la red.*

Normalmente en una red los enlaces son bidireccionales, es decir, un enlace une dos nodos en ambos sentidos.

Definición A.2. ***Enlace bidireccional****. Dados dos nodos,* $n_1 \in \mathscr{N}$ *y* $n_2 \in \mathscr{N}$*, en una red* $< \mathscr{N}, \mathscr{L} >$ *entonces se cumple que si* $l_{1,2} =< n_1, n_2 > \in \mathscr{L}$ *entonces también* $l_{2,1} =< n_2, n_1 > \in \mathscr{L}$.

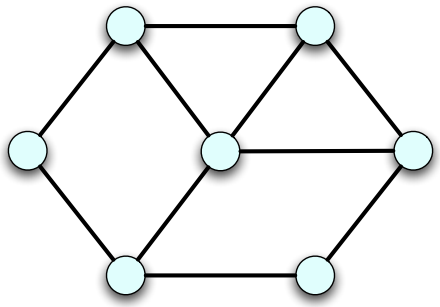


Figura A.1: Red lógica.

La topología lógica de una red es la que viene determinada por el conjunto $< \mathscr{N}, \mathscr{L} >$, por ejemplo la mostrada en la figura A.1. Sin embargo, en muchas ocasiones un fallo físico provoca el fallo en varios enlaces, debido a que estos pueden estar relacionados a nivel físico, tal y como se puede ver en la figura A.2. Es por esto que es conveniente poder relacionar enlaces entre sí.

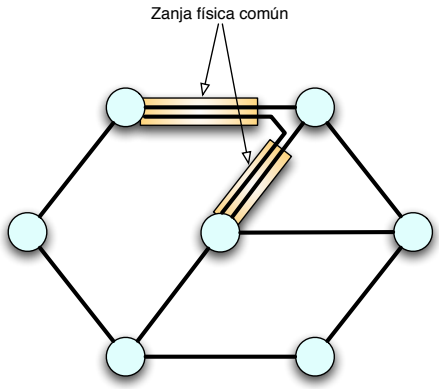


Figura A.2: Red física.

Para ello podemos definir un nuevo conjunto de enlaces con atributos, $\mathscr{L}'$, de modo que
$\mathscr{L}' =< \mathscr{L}, A >$
, donde A es el conjunto de atributos, tal que $A =< color, BW, ocupacion(t), SRLG, \ldots >$

Definición A.3. *La nueva definición de* ***red "física"*** *vendrá dada por*
$Red =< \mathscr{N}, \mathscr{L}' >$

De los conjuntos de atributos el más interesante es el SRLG.

Definición A.4. *SRLG es el identificador de un grupo de enlaces con el mismo riesgo [94].*

De modo que
$SRLG \subset \mathbb{N}$, números naturales que enumeran recursos físicos, tales como, zanjas, tramos de canalización, arquetas, etc atravesados por un enlace.

Es posible definir la función SRLG del siguiente modo:

Definición A.5. $SRLG(l_{1,2}) : \mathscr{L} \longrightarrow \mathbb{N}^*$ *devuelve los SRLGs contenidos en el enlace* l.

Nótese que este modelo de red (con SRLGs) soporta múltiples enlaces entre dos nodos. Como ejemplo véase la figura A.3.

Un camino es una secuencia ordenada de nodos y enlaces con atributos en una red de un nodo de la red a otro. Formalmente:

Definición A.6. *Dada la red* $N =< \mathscr{N}, \mathscr{L}' >$
, entonces $p(n_1, n_m) =< n_1, l_{1,2}, n_2, l_{2,3}, ..., l_{m-1,m}, n_m > / n_i \in \mathscr{N}, l_i \in \mathscr{L}$ y $p \in \mathscr{P}$
, siendo $\mathscr{P}$ *el conjunto de posibles caminos de N.*

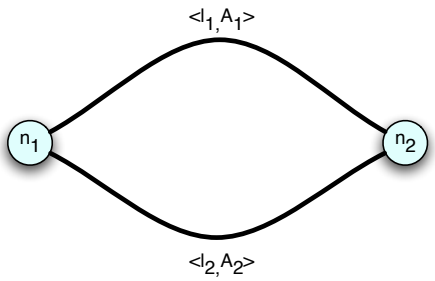


Figura A.3: Dos nodos unidos por más de un enlace.

Definición A.7. *$Nodos(p) : \mathscr{P} \longrightarrow \mathscr{N}^+$ devuelve el conjunto de nodos del camino p.*

Definición A.8. *$Enlaces(p) : \mathscr{P} \longrightarrow \mathscr{L}'^+$ devuelve el conjunto de enlaces del camino p.*

Definición A.9. *Dos caminos, p_1 y p_2, son disjuntos sii $\forall n_i \in Nodos(p_1)$ y $\forall l_i \in Enlaces(p_1)$ ocurre que $n_i \notin Nodos(p_2)$ y $l_i \notin Enlaces(p_2)$.*

Definición A.10. *El **camino principal** entre dos nodos de una red es el camino que seguirán los paquetes transmitidos desde el primero de ellos al segundo. En el caso de ser una red MPLS este camino estará formado por LSRs y coincidirá con el LSP principal construido en la red.*

Definición A.11. *El **camino de respaldo** entre dos nodos de una red es el camino que se utilizará en caso de fallo de algún elemento de red del camino principal. En el caso de ser una red MPLS este camino estará formado por LSRs y coincidirá con el LSP de respaldo construido en la red.*

Exiten dos tipos de redes *k*-conectadas, las redes con grafo de red *k*-nodo-conectadas y las redes con grafo de red *k*-enlace-conectadas.

Definición A.12. *Una red $\mathscr{R}$ con el conjunto de nodos $\mathscr{N}(\mathscr{R})$ se dice que es k-nodo-conectada si $\mathscr{N} \setminus X$ es conectada para todo $X \subseteq \mathscr{N}(\mathscr{R})$ con $|X| < k$.*

Es decir, una red es *k*-nodo-conectada si la red permanece conectada si se eliminan menos de *k* nodos de la red.

Definición A.13. *Una red $\mathscr{R}$ con el conjunto de enlaces $\mathscr{L}(\mathscr{R})$ se dice que es k-enlace-conectada si $\mathscr{R} \setminus X$ es conectado para todo $X \subseteq \mathscr{L}(\mathscr{R})$ con $|X| < k$.*

Es decir, una red es *k*-enlace-conectada si la red permanece conectada si se eliminan menos de *k* enlaces de la red.

Generalmente cuando no se especifica se entiende que se habla de una red k-nodo-conectada. Por ejemplo, en esta Tesis Doctoral cuando se refiere a una red 2-nodo-conectada se utiliza simplemente el termino red 2-conectada. En esta Tesis Doctoral, y en general en literatura matemática, una red 2-nodo-conectada, 2-conectada o biconectada es el mismo tipo de red.

Lema A.1. *Si $k(\mathscr{R})$ es el grado de nodo-conectividad de la red $\mathscr{R}$ y $k'(\mathscr{R})$ es el grado de enlace-conectividad de la red $\mathscr{R}$ se cumple siempre que $k(\mathscr{R}) \leq k'(\mathscr{R})$ para cualquier $\mathscr{R}$.*

Teorema A.1. *Además, si lambda$(\mathscr{R})$ es el mínimo número de caminos independientes (disjuntos) entre dos nodos cualesquiera de la red $\mathscr{R}$, entonces $k(\mathscr{R}) = \lambda(\mathscr{R})$ (teorema de Menger).*

Una red 2-conectada, por tanto, es una red en la que existen al menos 2 caminos disjuntos entre cada par de nodos. Retirar (que haya un fallo en) un nodo o un enlace de la red no la desconecta, con lo que sigue existiendo un camino que conecta cualquier par de nodos.

Apéndice B

Objetos RSVP-TE propuestos

B.1. Objeto IDRO

En este apéndice se va a especificar el formato del objeto IDRO (*Inter-Domain Routing Object*) propuesto en esta Tesis Doctoral como parte de la solución del mecanismo propuesto en el capítulo 6.

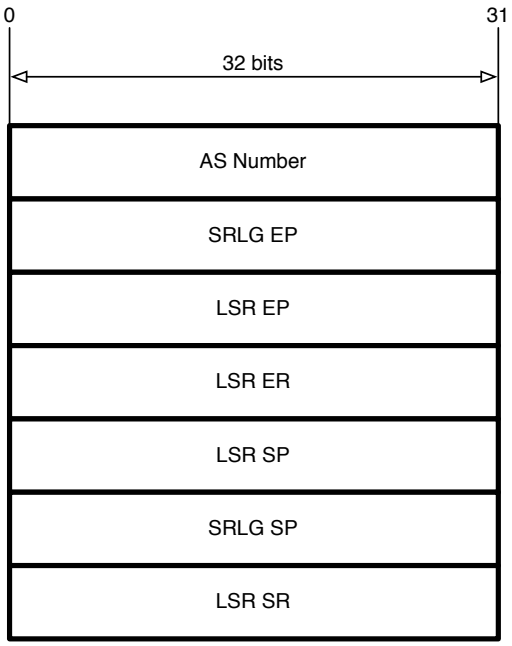


Figura B.1: Especificación del objeto RSVP de tipo IDRO.

El objeto IDRO está formado por una tupla de siete elementos. Estos son; el número de AS, el SRLG de entrada del LSP principal, el LSR de entrada del LSP

principal, el LSR de entrada del LSP de respaldo, el LSR de salida del LSP principal, el SRLG de salida del LSP de respaldo y el LSR de salida del LSP de respaldo. Como se puede ver hay tres tipos de subobjetos diferentes, número AS, SRLGs y LSRs. La figura B.1 muestra un esquema del objeto.

Los subobjetos *SRLG EP*, *LSR EP*, *LSR SP* y *SRLG SP* del objeto IDRO deben ser incluidos en un objeto XRO, por tanto deben cumplir las especificaciones de los subobjetos de tipo XRO. Los subobjetos *AS number*, *LSR ER* y *LSR SR* pueden incluirse en un objeto de tipo ERO, por lo que deberán respetar el formato de los subobjetos de tipo ERO.

Cada uno de los subobjetos presentes en el objeto IDRO está definido en diversas RFCs. Debido a que estos subobjetos pueden ser utilizados posteriormente en objetos de tipo XRO o ERO se utilizarán los subobjetos tal y como están definidos para estos objetos RSVP-TE.

Para todos los subobjetos que se van a definir el valor del bit L puede significar dos cosas.

Si el destino es un objeto de tipo ERO entonces, si el bit L está activado indica que el objeto representa un salto de tipo "*loose*". Si el bit L no está activado indica que el objeto representa un salto de tipo "*strict*" [88].

En cambio si el subobjeto destino es de tipo XRO entonces, si el bit L está activo significa que el nodo debe ser evitado. Si el bit L no está activo implica que el nodo tiene que ser excluido [95].

AS number El subojeto *AS number* viene especificado en la RFC 3209 [88] como un nodo abstracto del objeto ERO. En la figura B.2 se puede ver la disposición de los campos:

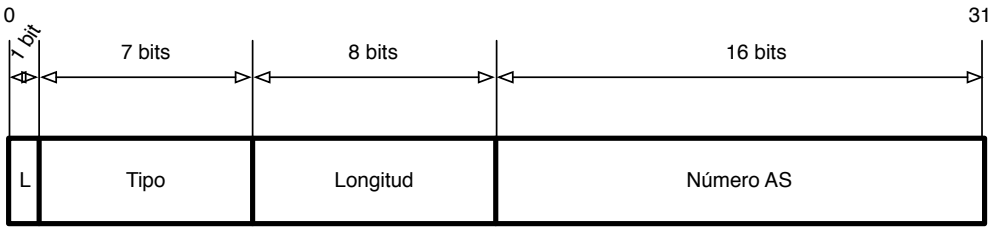


Figura B.2: Subobjeto *AS number*

- Tipo: 32
- Longitud: 4

- *AS number*: Número de Sistema Autónomo. Es un nodo abstracto que representa al conjunto de nodos que pertenece al Sistema Autónomo.

SRLG El subobjeto SRLG viene definido en la RFC 4874 [95] como un subobjeto del objeto XRO pero modelado según el modelo de subobjetos de tipo ERO definidos en la RFC 3209 [88]. En la figura B.3 se puede ver la disposición de los campos:

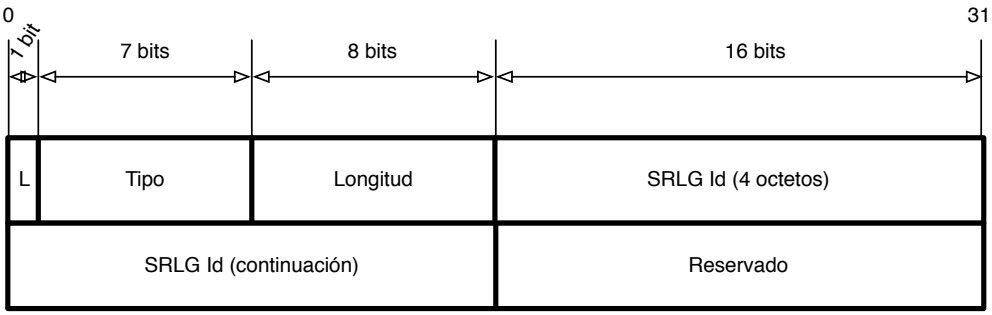


Figura B.3: Subobjeto SRLG

- Tipo: 34
- Longitud: 8
- SRLG Id: Identificador de 32 bits del SRLG.
- Reservado: Este campo está reservado. Debe estar puesto a cero al transmitirlo y tiene que ser ignorado en la recepción.

Subobjeto LSR El objeto LSR representa a un nodo o un conjunto de nodos. Existen dos posibles subobjetos, un prefijo IPv4 o un prefijo IPv6.

Los subobjetos prefijo IPv4 y prefijo IPv6 vienen definidos en la RFC 3209 [88] como un subobjeto del objeto ERO y RRO, también se han redefinido en la RFC 4874 [95] para el objeto XRO. La diferencia entre ambos, además del significado del bit L es el último campo, que bien puede ser un campo de atributos (objeto XRO) o de flags (objeto ERO). En función del objeto destino se utilizará un significado u otro. En estas RFCs se muestran los distintos significados de este campo.

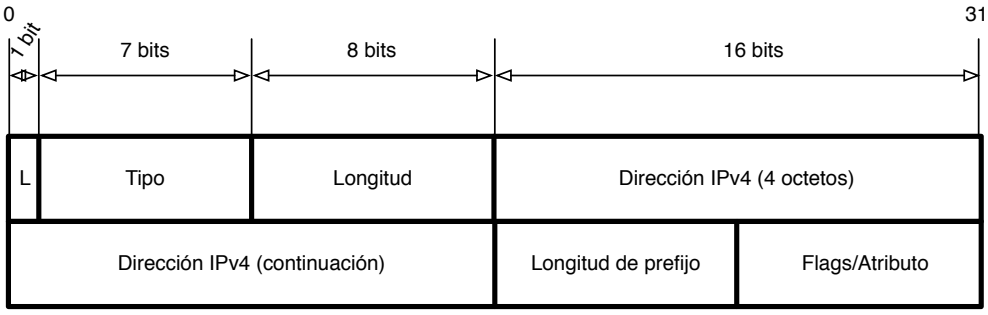


Figura B.4: Subobjeto IPv4

Prefijo IPv4 En la figura B.4 se muestra un esquema de este subobjeto. El significado de cada campo es el siguiente:

- Tipo: 1
- Longitud: 8
- Dirección IPv4: Una dirección unicast de 32 bits.
- Longitud del prefijo: Prefijo a tener en cuenta de la dirección.

Prefijo IPv6 En la figura B.4 se muestra un esquema de este subobjeto. El significado de cada campo es el siguiente:

- Tipo: 2
- Longitud: 20
- Dirección IPv6: Una dirección unicast de 128 bits.
- Longitud del prefijo: Prefijo a tener en cuenta de la dirección.

B.2. Objeto DPMMO

En este apéndice se va a especificar el formato del objeto DP1MO (*Disjoint Paths One Origin Multiples Destinations Object*) propuesto en esta Tesis Doctoral como parte de la solución del mecanismo propuesto en el capítulo 5.

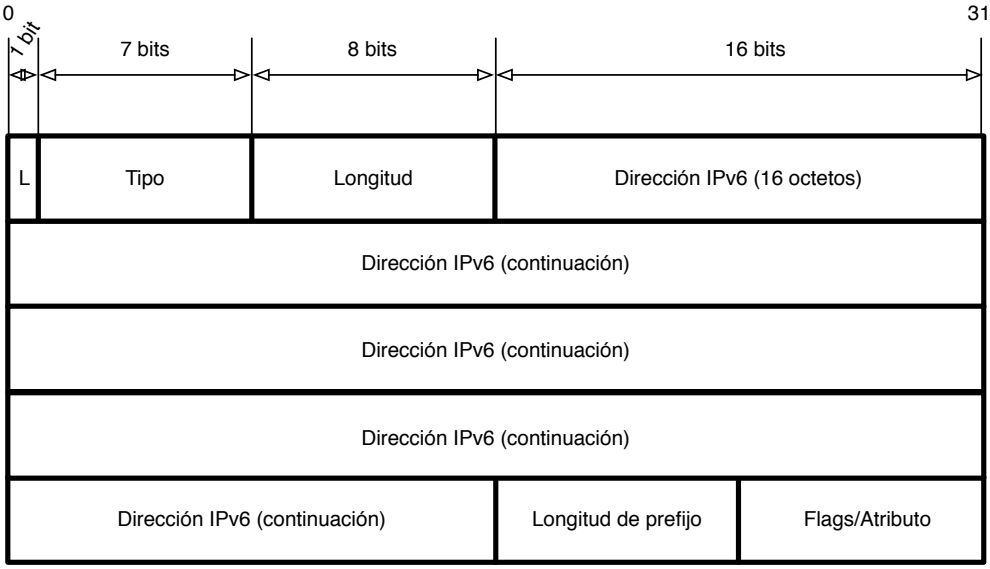


Figura B.5: Subobjeto IPv6

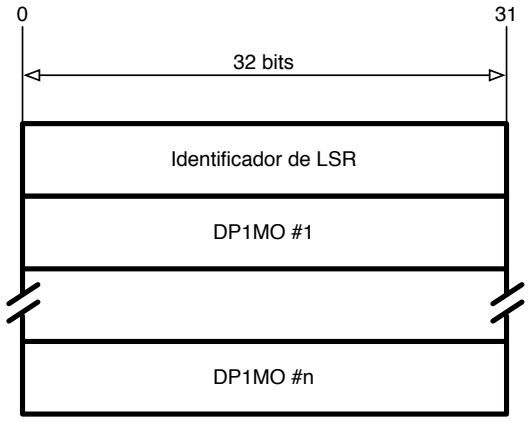


Figura B.6: Especificación del objeto RSVP de tipo DP1MO.

El objeto está formado por un identificador de LSR, que identifica el LSR que ha generado el objeto y una serie de subobjetos DP1MO.

El ***identificador de LSR*** es un subobjeto de tipo prefijo IPv4 o prefijo IPv6. Sus formatos están definidos en el apéndice B.1. Todos los campos de estos dos subobjetos se definen igual que en el caso del apéndice B.1 excepto el campo Flags/Atributo que aquí adquiere un significado distinto y el bit L que no aplica.

El campo de Flags/Atributo, ahora *número de orígenes*, es un número entero que indica cuantos subobjetos DP1MO contiene el objeto.

A continuación especificaremos el subobjeto DP1MO.

B.2.1. El subobjeto DPOMO

El subobjeto DP1MO está formado por diversos campos mostrados en la figura B.7 y que detallaremos a continuación.

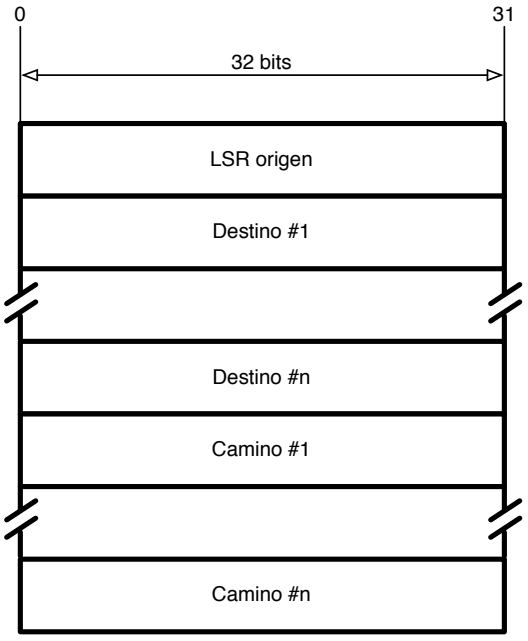


Figura B.7: Especificación de un subobjeto de DP1MO

El ***LSR de origen*** y cada uno de los ***destinos*** es un subobjeto de tipo prefijo IPv4 o prefijo IPv6. Son los LSRs de los que se van a especificar los caminos disjuntos, desde el *origen* a los diferentes *destinos*.

Es este caso, el campo Flags/Atributo adquiere un significado especial dependiendo de si se corresponde con un *origen* o con un *destinos*.

En el caso del *origen*, el campo se trata de ***número de destinos***, y se corresponde con un número entero que indica cuantos destinos viene a continuación.

En el caso de los *destinos*, el campo se trata de ***longitud del camino***, y se corresponde con un número entero que indica el número de elementos incluidos en el elemento ***camino*** correspondiente a ese destino. Si este valor es cero es que no se va a especificar el camino para ese destino. De este modo, este mismo objeto es válido para ser utilizado cuando se quiere ocultar la información interior y sólo se facilitan los LSR de entrada y salida.

Cada uno de los ***caminos*** es una lista de subobjetos de tipo prefijo IPv4, prefijo IPv6 o SRLG que indican el camino desde el origen al destino. Los subobjetos son los mismos especificados en el apéndice B.1. Estos dos nodos, origen y destino, no es necesario ponerlos, puesto que se conoce cuáles son.

B.3. Objeto IDLO

En este apéndice se va a especificar el objeto IDLO (*Inter-Domain LSRs Object*) propuesto en esta Tesis Doctoral como parte de la solución del mecanismo propuesto en el capítulo 5.

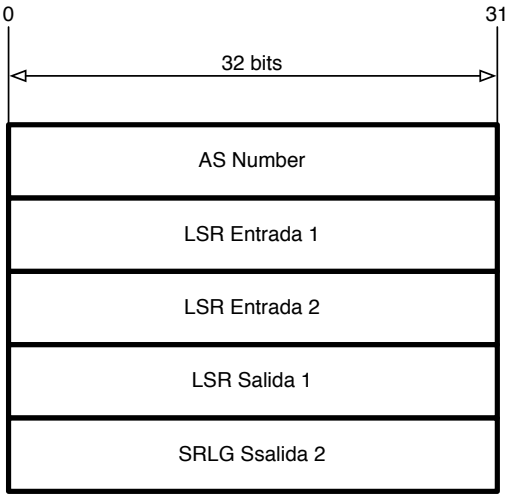


Figura B.8: Especificación del objeto RSVP de tipo IDLO

Como puede comprobarse en la figura B.8 el objeto IDLO está formado por 5 subobjetos, el número de AS, y cuatro LSRs, dos de entrada y dos de salida.

El subobjeto *AS number* se corresponde con el definido en el apéndice B.1. Los otros cuatro subojetos, de tipo LSR, *LSR de Entrada 1*, *LSR de Entrada 2*, *LSR*

de Salida 1 y *LSR de Salida 2*, pueden ser de tipo prefijo IPv4 o prefijo IPv6. Sus formatos están también definidos en el apéndice B.1.

LSR de Entrada 1 y *LSR de Entrada 2* son los LSR de entrada del dominio *AS number* que se utilizarán como parte de los LSPs, principal y de respaldo, que se quieren utilizar. No se especifica cuál es principal y cuál es de respaldo porque depende del dominio anterior, de a dónde acceda el LSP principal que se está construyendo desde dominios anteriores.

Lista de Acrónimos

APD Algoritmo de las Parejas Disjuntas, 43–46, 49, 56, 57, 133
ARO Associated Route Object, 22–25, 32, 61, 62
AS Autonomous System, 5, 21, 26, 27, 29, 90, 92
ATM Asynchronous Transfer Mode, 9, 12

BGP Border Gateway Protocol, 7, 11, 26, 27, 29–31, 34, 90–92, 94–96, 119, 127, 131, 134

CbR Constraint-based Routing, 30
CE Customer Edge Router, 11

DiffServ Differentiated Services, 12, 13
DPMMO Disjoint Paths Multiples Origins Multiples Destinations Object, 63, 65, 71
DP1MO Disjoint Paths One Origin Multiples Destinations Object, 144, 146
DSCP Differentiated Services Code Point, 12
DTMF Dual Tone Multi-Frequency, 10

EGP Exterior Gateway Protocol, 30, 31
E-LSP EXP-Inferred-PSC LSP, 12
ERO Explicit Route Object, 24, 85, 110, 142
EXP Experimental bits, 12
EXRS Explicit eXclusion Route Subobject, 24

FEC Forwarding Equivalency Class, 9
FIB Forwarding Information Base, 95

GMPLS Generalised MPLS, 6, 20, 23

IBGP Interior BGP, 69, 92
IBLBT InterDomain Boundary Local Bypass Tunnel, 19
IDLO Inter-Domain LSRs Object, 66, 147
IDR Inter-Domain Routing, 7
IDRO Inter-Domain Routing Object, 73, 83–86, 141

IETF Internet Engineering Task Force, 6, 7, 11, 19, 20, 23, 24
IGP Interior Gateway Protocol, 22, 23, 30, 31, 37, 40, 71, 111
IntServ Integrated Services, 12
IP Internet Protocol, 5, 7, 9–12, 18, 24, 30, 34
IPSec Internet Protocol Security, 12
IS-IS Intermediate System to Intermediate System, 4, 6, 39

LAN Local Area Network, 7
LDP Label Distribution Protocol, 11
L-LSP Label-Only-Inferred-PSC LSP, 12
LSA Link State Advertisement, 71
LSP Label Switched Path, 6, 9, 17–21, 26, 110
LSR Label Switching Router, 9, 10, 13–15, 17, 18

MPLS Multiprotocol Label Switching, 5–7, 9–20, 30, 34, 110, 127

NHOP Next Hop, 16, 17
NLRI Network Layer Reachability Information, 11, 92
NNHOP Next-Next Hop, 16, 17

OAM Operations and Management, 6
OIF Optical Internetworking Forum, 19
OSPF Open Shortest Path First, 4, 6, 11, 31, 39, 67, 70, 71

PCC Path Computation Client, 6
PCE Path Computation Element, 5, 6, 20, 21, 27, 32–34, 37, 43, 61, 113, 119, 120
PE Provider Edge Router, 11
PHB Per Hop Behaviour, 12
PHP Penultimate Hop Popping, 16
P-LSA Path-LSA, 71
PML Path Merge LSR, 13, 14, 16, 17, 75
PPRO Primary Path Route Object, 5, 22–25, 32, 61, 62
PSL Path Switch LSR, 13–16, 18, 75, 112
P2MP Point to Multipoint, 6

QoS Quality Of Service, 12, 13, 33, 46, 47, 53, 131

RFC Request For Comment, 6, 10, 11, 14–17
RIB Route Information Base, 91, 92, 95–97
RIP Routing Information Protocol, 4
R-LSA Router-LSA, 71
RRO Record Route Object, 22, 63

RSVP Resource Reservation Protocol, 12, 22–24, 71
RSVP-TE RSVP Traffic Engineering, 11, 17, 26, 63, 66, 70, 71, 73, 83–86, 110, 111, 142

SIDR Secure Inter-Domain Routing, 7
SPT Shortest-Path Tree, 67
SRLG Shared Risk Link Group, 22, 24, 33, 43, 84, 138

TCAM Ternary Content Addressable Memory, 10
TLV Type Length Value, 11

VoIP Voice over IP, 10
VoMPLS Voice over MPLS, 10
VPLS Virtual Private LAN Service, 7, 10, 11
VPN Virtual Private Network, 7, 11, 12

WG Working Group, 20

XRO eXclude Route Object, 24, 85, 142

Bibliografía

[1] A. Fumagalli and L. Valcarenghi. IP Restauration vs. WDM Protection: Is there an Optimal Choice? *IEEE Network*, 14(6):34–41, November/December 2000.

[2] Marko Lacković and Robert Inkret. Overview of Resilience Schemes in Photonic Transmission Network. In *ICTON*. 7th International Conference on Transparent Optical Networks, July 3–7 2005.

[3] G. Malkin. RIP Version 2. RFC 2453 (Standard), November 1998. Updated by RFC 4822.

[4] J. Moy. OSPF Version 2. RFC 2328 (Standard), April 1998.

[5] R.W. Callon. Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195 (Proposed Standard), December 1990. Updated by RFC 1349.

[6] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. In *SIGCOMM*, pages 175–187, 2000.

[7] International Engineering Consortium. *Multiprotocol Label Switching (MPLS)*, 2007.

[8] The MPLS Resource Center. http://www.mplsrc.com/.

[9] J.W. Suurballe. Disjoint Paths in a Network. *Networks*, 4:125–145, June 1974.

[10] Fabio Ricciato, Ugo Monaco, and Daniele Alì. Distributed Schemes for Diverse Path Computation in Multidomain MPLS Networks. *IEEE Communications Magazine*, 43(6), June 2005.

[11] Marcelo Yannuzzi, Xavi Masip-Bruin, Sergio Sánchez, Jordi Domingo-Pascual, Ariel Orda, and Alex Sprintson. On the Challenges of Establishing Disjoint QoS IP/MPLS Paths Across Multiple Domains. *IEEE Communications Magazine*, pages 60–66, December 2006.

[12] J.P. Lang, Y. Rekhter, and D. Papadimitriou. RSVP-TE Extensions in support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery. *draft-ietf-ccamp-gmpls-recovery-e2e-signaling-04.txt*, October 2006. work in progress.

[13] A. Farrel, J.-P. Vasseur, and J. Ash. A Path Computation Element (PCE)-Based Architecture. RFC 4655 (Informational), August 2006.

[14] Multiprotocol Label Switching (mpls) Working Group. http://www.ietf.org/html.charters/mpls-charter.html.

[15] Common Control and Measurement Plane (ccamp) Working Group. http://www.ietf.org/html.charters/ccamp-charter.html.

[16] Path Computation Element (pce) Working Group. http://www.ietf.org/html.charters/pce-charter.html.

[17] Layer 2 Virtual Private Networks (l2vpn) Working Group. http://www.ietf.org/html.charters/l2vpn-charter.html.

[18] Layer 3 Virtual Private Networks (l3vpn) Working Group. http://www.ietf.org/html.charters/l3vpn-charter.html.

[19] Inter-Domain Routing (idr) Working Group. http://www.ietf.org/html.charters/idr-charter.html.

[20] Secure Inter-Domain Routing (idr) Working Group. http://www.ietf.org/html.charters/sidr-charter.html.

[21] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001.

[22] S. Harnedy. *The MPLS Primer*. Prentice Hall, Nov 2001.

[23] Anthony J. McAuley and Paul Francis. Fast Routing Table Lookup Using CAMs. In *INFOCOM (3)*, pages 1382–1391, 1993.

[24] Rajendra Persaud, Dirk Sabath, Gerald Berghoff, and Ralf Schanko. Performance Evaluation of MPLS and IP on an IXP1200 Network Processor. In *AINA (2)*, volume 2 of *ISSN: 1550-445X*, pages 413–417, April 2006. D.O.I. 10.1109/AINA.2006.268.

[25] Erik J. Johnson and Aaron R. Kunze. *IXP1200 Programming*. Intel Press, 2002.

[26] J. Ash, B. Goode, J. Hand, and R. Zhang. Requirements for Header Compression over MPLS. RFC 4247 (Informational), November 2005.

[27] MPLS Forum Technical Committee. Voice over MPLS - Bearer Transport Implementation Agreement. http://www.mfaforum.org/tech/VoMPLS_IA.pdf, July 27 2001.

[28] VPLS ORG. http://vpls.org/.

[29] L. Martini, Ed., E. Rosen, N. El-Aawar, and G. Heron. Encapsulation Methods for Transport of Ethernet over MPLS Networks. RFC 4448 (Proposed Standard), April 2006.

[30] MPLS and Frame Relay Alliance (MFA) FORUM. http://www.mfaforum.org.

[31] K. Kompella and Y. Rekhter. Virtual Private LAN Service (VPLS) Using BGP for Auto-discovery and Signaling. Internet Draft, June 2006. Trabajo en desarrollo.

[32] Marc Lasserre and Vach Kompella. Virtual Private LAN Services Using LDP. Internet Draft, June 2006. Trabajo en desarrollo.

[33] Ximing Dong and Shaohua Yu. An Iterative Algorithm in Building Delay-constrained Multicast Trees over VPLS Domain. In *ICN/ICONS/MCL*, page 146, 2006.

[34] Ximing Dong and Shaohua Yu. Deliver Multicast Traffic over VPLS Domain Using Aggregated Multicast Trees. In *AICT-ICIW '06: Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services*, page 59, Washington, DC, USA, 2006. IEEE Computer Society.

[35] N.A. Ali, H. Mouftah, and S. Gazor. A distributed bandwidth-guaranteed routing algorithm for point-to-multipoint VPLS virtual connections. In *Conference on Communications, 2005. ICC 2005. 2005 IEEE International*, volume 3, pages 1561–1565. IEEE Computer Society, May 2005.

[36] Najah AbuAli, Hussein T. Mouftah, and Saeed Gazor. Multi-Path Traffic Engineering Distributed VPLS Routing Algorithm. In *ICW/ICHSN/ICMCS/SENET*, pages 275–280, 2005.

[37] N.A. Ali, H. Mouftah, and S. Gazor. QoS guarantees of point-to-multipoint VPLS connections. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 3, page 5. IEEE Computer Society, Nov 2005.

[38] R. Aggarwal and D. Papadimitriou. Extensions to RSVP-TE for Point to Multipoint TE LSPs. Internet Draft, May 2006. Trabajo en desarrollo.

[39] K. Moerman, J. Fishburn, M. Lasserre, and D. Ginsburg. Utah's UTOPIA: an ethernet-based MPLS/VPLS triple play deployment. *Communications Magazine, IEEE*, 43(11):142–150, 2005.

[40] E. Rosen and Y. Rekhter. BGP/MPLS VPNs. RFC 2547 (Informational), March 1999. Obsoleted by RFC 4364.

[41] E. Rosen and Y. Rekhter. BGP/MPLS IP Virtual Private Networks (VPNs). RFC 4364 (Proposed Standard), February 2006. Updated by RFCs 4577, 4684.

[42] E. Rosen, P. Psenak, and P. Pillay-Esnault. OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs). RFC 4577 (Proposed Standard), June 2006.

[43] C. Siriakkarap, P. Setthawong, and S. Tanterdtid. RSVP Based Critical Services Path Recovery in MPLS Network. Information and Telecommunication Technologies, 2005. APSITT 2005 Proceedings. 6th, Nov 2005.

[44] Cisco Systems, Inc. *Analysis of MPLS-Based IP VPN Security: Comparison to Traditional L2VPNs Such as ATM and Frame Relay, and Deployment Guidelines*, 2004.

[45] Rong Ren, Deng-Guo Feng, and Ke Ma. A detailed implement and analysis of MPLS VPN based on IPSec. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, volume 5, pages 2779 – 2783, August 2004.

[46] Eric C. Rosen, Jeremy De Clercq, Olivier Paridaens, Yves T'Joens, and Chandru Sargor. Architecture for the Use of PE-PE IPsec Tunnels in BGP/MPLS IP VPNs. Internet Draft, August 2005. Trabajo en desarrollo.

[47] Nicolas Rouhana and Eric Horlait. Differentiated Services and Integrated Services use of MPLS. In *Proceedings ISCC 2000*. Fifth IEEE Symposium on Computers and Communications, 2000.

[48] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen. Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. RFC 3270 (Proposed Standard), May 2002.

[49] S. Avallone, M. Esposito, A. Pescape, and G. Romano, S.P. Ventre. An experimental analysis of diffserv-mpls interoperability. International Conference on Telecommunications, 2003. ICT 2003, 23 Feb.-1 March 2003.

[50] Wei-Chu Lai, Kuo-Ching Wu, and Ting-Chao Hou. Design and Evaluation of Diffserv Functionalities in the MPLS Edge Router Architecture. In *First International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications (AAA-IDEA'05)*, pages 18–25, 2005.

[51] V. Jaiswal and S. Kumar. Intel IXP28XX network processor based NG Edge Router. In *First International Conference on Communication System Software and Middleware, 2006. Comsware 2006*, pages 1 – 12, 08-12 January 2006.

[52] Jean-Philippe Vasseur, Mario Pickavet, and Piet Demeester. *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[53] V. Sharma and F. Hellstrand. Framework for Multi-Protocol Label Switching (MPLS)-based Recovery. RFC 3469 (Informational), February 2003.

[54] P. Pan, G. Swallow, and A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090 (Proposed Standard), May 2005.

[55] Svetlin Petrov. Mpsl traffic protection. In *International Conference on Computer Systems and Technologies - CompSysTech' 2006*, 2006.

[56] S.Makam, V.Sharma, K.Owens, and C.Huang. Protection/Restoration of MPLS Networks. draft-makam-mpls-protection-00.txt, October 1999.

[57] D. Haskin and R.Krishnan. A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute. draft-haskin-mpls-fast-reroute-05.txt, November 2000.

[58] C. Huang, V. Sharma, K. Owens, and S.Makam. Building Reliable MPLS Networks Using a Path Protection Mechanism. *IEEE Communications Magazine*, March 2002.

[59] Yehuda Afek, Anat Bremler-Barr, Haim Kaplan, Edith Cohen, and Michael Merritt. Restoration by path concatenation: fast recovery of MPLS paths. *Distrib. Comput.*, 15(4):273–283, 2002.

[60] W. D. Grover and D. Stamatelakis. Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration. In *IEEE International Conference on Communications (ICC) '98*, Atlanta, Georgia, USA, 7-11 Jun 1998.

[61] D. Stamatelakis and W. Grover. IP Layer Restoration and Network Planning Based on Virtual Protection Cycles. *IEEE JSAC*, 18(10), October 2000.

[62] J. Doucette, P.A. Giese, and W.D. Grover. Combined node and span protection strategies with node-encircling p-cycles. *Proceedings.5th International Workshop on Design of Reliable Communication Networks, 2005. (DRCN 2005)*, 16-19 October 2005.

[63] Wayne D. Grover and Gangxiang Shen. Extending the p-Cycle Concept to Path-Segment Protection. In *ICC2003*, 2003.

[64] C. Gruber and D. Schupke. Capacity-efficient planning of resilient networks with p-cycles. In *10th International Telecommuniation Network Strategy and Planning Symposium*, 2002.

[65] D. Schupke, C. Gruber, and A. Autenrieth. Optimal Configuration of p–Cycles in WDM Networks. In IEEE, editor, *ICC 2002*, 2002.

[66] Otto Wittner, Bjarne E. Helvik, and Victor Nicola. Link and node protection using hamiltonian p-cycles found by ant-like agents. In *1st euroNGI Workshop on Traffic Engineering, Protection and Restoration for NGI*. Lund, Sweden, May 27-28 2004.

[67] C. G. Gruber. Resilient Networks with Non-Simple p-Cycles. In *ICT 2003*, volume 2, pages 1027 – 1032, Feb-March 2003.

[68] A. Farrel, J.-P. Vasseur, and A. Ayyangar. A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering. RFC 4726 (Informational), November 2006.

[69] Jean-Philippe Vasseur and Arthi Ayyangar. Inter-area and Inter-AS MPLS Traffic Engineering. Internet Draft, February 2004.

[70] J.-L. Le Roux, J.-P. Vasseur, and J. Boyle. Requirements for Inter-Area MPLS Traffic Engineering. RFC 4105 (Informational), June 2005.

[71] Angela L. and John Strand. Control Plane Considerations for All-Optical and Multi-Domain Optical Networks and Their Status in OIF and IETF. Technical report, Celion Networks and AT&T Optical Networks Research Dept., November 2001.

[72] Changcheng Huang and Donald Messier. A Fast and Scalable Inter-Domain MPLS Protection Mechanism. *Journal of Communications and Networks*, 6(1), March 2004.

[73] A. Farkas, J. Szigeti, and T. Cinkler. P-cycle based protection schemes for multi-domain networks. In *Design of Reliable Communication Networks (DRCN05)*, 2005.

[74] WGN5: V Workshop in G/MPLS networks. *Combining Border Router Policies for Disjoint LSP computation*, 30-31 March 2006.

[75] A. D'Achille, M. Listanti, U. Monaco, F. Ricciato, D. Ali, and V. Sharma. Diverse Inter-Region Path Setup/Establishment. *draft-dachille-diverse-inter-region-path-setup-01.txt*, October 2004.

[76] Cisco Systems, Inc. *Guide to ATM Technology. For the Catalyst 8540 MSR, Catalyst 8510 MSR, and LightStream 1010 ATM Switch Routers*, 1999-2000.

[77] CY. Lee, A. Farrel, and S. De Cnodder. Exclude Routes - Extension to RSVP-TE. *draft-ietf-ccamp-rsvp-te-exclude-route-06.txt*, November 2006. work in progress.

[78] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE. *draft-ietf-ccamp-crankback-06.txt*, January 2007. work in progress.

[79] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine*, 2003.

[80] WGN4: IV Workshop in G/MPLS networks. *MPLS-supported interdomain recovery in the public Internet*, March 2005.

[81] David Larrabeiti, Ricardo Romeral, Ignacio Soto, Manuel Urueña, Tibor Cinkler, János Szigeti, and János Tapolcai. MultiDomain Issues of Resilience. In *7th International Conference on Transparent Optical Networks*, July 3-7 2005.

[82] Ricardo Romeral, Marcelo Yannuzzi, David Larrabeiti, Xavier Masip-Bruin, and Manuel Urueña. Multi-domain G/MPLS recovery paths using PCE. In *10th European Conference on Networks & Optical Communications*, 2005.

[83] Sukrit Dasgupta, Jaudelice C. de Oliveira, and Jean-Philippe Vasseur. Path-Computation-Element-Based Architecture for Interdomain MPLS/GMPLS Traffic Engineering: Overview and Performance. *IEEE Network*, 21(4):38–45, July 2007.

[84] A. Pattavina. *Switching Theory*. Wiley, 1998.

[85] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771 (Draft Standard), March 1995. Obsoleted by RFC 4271.

[86] Timothy G. Griffin and Gordon T. Wilfong. An Analysis of BGP Convergence Properties. In *Proceedings of SIGCOMM*, pages 277–288, Cambridge, MA, August 1999.

[87] Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent Route Oscillations in Inter-Domain Routing. *Computer Networks*, 32(1):1–16, 2000.

[88] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels. RFC 3209 (Proposed Standard), December 2001. Updated by RFCs 3936, 4420, 4874.

[89] Iljitsch van Beijnum. *BGP: Building Reliable Networks with the Border Gateway Protocol*. O'Reilly, 2002.

[90] Eusebi Calle, José L. Marzo, and Anna Urra. Protection Performance Components in MPLS Networks. *Computer Communications*, 27(12):1220–1228, July 2004.

[91] János Szigeti, Ricardo Romeral, Tibor Cinkler, and David Larrabeiti. P-Cycle Protection in Multi-Domain Environment. Pendiente de publicación.

[92] Wayne D. Grover. Understanding p-Cycles, Enhanced Rings, and Oriented Cycle Covers. In *1st Int'l Conference on Optical Communications and Networks (ICOCN'02)*, pages 305–308, 11-14 Noviembre 2002.

[93] W. D. Grover and D. Stamatelakis. Bridging the ring- mesh dichotomy with p-cycles. In *Design of Reliable Communication Networks (DRCN 2000)*, 9-12 Abril 2000.

[94] Sudheer Dharanikota, Raj Jain, Riad Hartani, Dimitri Papadimitriou, Yong Xue, and Curtis Brownmiller. On Shared Risk Link Groups for diversity and risk assessment, March 2001.

[95] CY. Lee, A. Farrel, and S. De Cnodder. Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE). RFC 4874 (Proposed Standard), April 2007.

[96] Ricardo Romeral, Dimitri Staessens, David Larrabeiti, Mario Pickavet, and Piet Demeester. End-to-end Survivable Connections in Multi-Domain GMPLS Networks. In Pere Vilà and Eusebi Calle, editors, *VI Workshop in G/MPLS Networks*, ISBN 978-84-96742-20-8, pages 75–84, Gerona, España, 12-13 Abril 2007. Documenta Universitaria.

[97] Ricardo Romeral and David Larrabeiti. Combining border router policies for disjoint LSP computation. In José L. Marzo, editor, *V Workshop in G/MPLS Networks*, ISBN 84-934349-0-6, pages 181–190, Gerona, España, 30-31 Marzo 2006. Documenta Universitaria.

[98] David Larrabeiti, Ricardo Romeral, Ignacio Soto, Manuel Urueña, Tibor Cinkler, János Szigeti, and János Tapolcai. Multi-Domain Issues of Resilience. In Marian Marcinak, editor, *7th International Conference on Transparent Optical Networks*, ISBN 0-7803-9236, pages 103–114, Barcelona, España, 2005. IEEE.

[99] Ricardo Romeral, Marcelo Yannuzzi, David Larrabeiti, Xavier Masip-Bruin, and Manuel Urueña. Multi-domain G/MPLS recovery paths using PCE. In *Proceedings of the 10th European Conference on Networks and Optical Communications*, ISBN 0-9538863-8-7 5-7, pages 187–193, London, UK, 5 Julio 2005. University College London.

[100] Ricardo Romeral, David Larrabeiti, Miguel Couto, Macelo Bagnulo, and Aberto García. MPLS-supported interdomain recovery in the public Internet. In Teodor Jové and Ramon Fabregat, editors, *IV Workshop in G/MPLS Networks*, ISBN 84-933783-4-8, pages 103–114, Gerona, España, 21–22 Abril 2005. Documenta Universitaria.