TESIS DOCTORAL

"Gestión de la pérdida de datos en redes PTN para servicios con requisitos de QoS"

Francisco Javier Rodríguez Pérez

Departamento de Ingeniería de Sistemas Informáticos y Telemáticos

Conformidad del Director:

Fdo.:

Año de lectura: 2015

TESIS DOCTORAL

Gestión de la pérdida de datos en redes PTN para servicios con requisitos de QoS

Departamento de Ingeniería de Sistemas Informáticos y Telemáticos

Francisco Javier Rodríguez Pérez

Año 2015

Gestión de la pérdida de datos en redes PTN para servicios con requisitos de QoS

TESIS DOCTORAL

Francisco Javier Rodríguez Pérez

Director:

Dr. D. José Luis González Sánchez

Línea de Investigación:
Computación y Comunicaciones de Altas Prestaciones

Grupo de Investigación:
Grupo de investigación de Ingeniería Telemática Aplicada y Comunicaciones Avanzadas (GÍTACA)

Departamento de Ingeniería de Sistemas Informáticos y Telemáticos
Escuela Politécnica
Universidad de Extremadura

Cáceres, Spain. Año 2015

A mi familia, que me ha acompañado a lo largo de toda esta aventura.

A Noemi, que me alentó a continuar en los momentos más difíciles.

A mi director y a mis compañeros de GÍTACA, que me apoyaron para concluirla.

Resumen

La aparición de nuevos tipos de aplicaciones, como los servicios de transferencia masiva de datos, la virtualización de servidores, *cloud computing* o las recientes propuestas relacionadas con *Big Data*, ha conducido a un uso mucho mayor de los recursos disponibles en las redes multi-propósito de Internet. Estos servicios están dando lugar a un evidente incremento del consumo del ancho de banda en momentos puntuales, provocando congestiones inesperadas, especialmente en las redes troncales, donde coexisten flujos de datos de muy diverso ámbito. Estas variaciones dinámicas de la carga de la red hacen que siga siendo razonable la mejora en el aprovisionamiento de recursos de infraestructura de red, junto con el consiguiente aumento de capacidad. Sin embargo, esto no ha sido suficiente a la hora de ofrecer la *Quality of Service* (QoS) requerida por los nuevos tipos de aplicaciones emergentes. Para el caso de redes en explotación basadas en el protocolo *Internet Protocol* (IP), la incorporación de mecanismos efectivos de control de la congestión ha sido un problema difícil de resolver, debido a la limitada capacidad de la propia tecnología IP. En principio, *Multiprotocol Label Switching Transport Profile* (MPLS-TP) es una tecnología de intercambio de etiquetas orientada a conexión ofreciendo nuevas posibilidades a la hora de abordar estas limitaciones. Permite al operador de servicios emplear sofisticados mecanismos de control del rendimiento del tráfico, de cara a las redes de nueva generación orientadas a conexión y de conmutación de paquetes. Sin embargo, no ofrece, de forma nativa, mecanismos que afronten la necesidad de mejora del rendimiento y eficiencia de las actuales técnicas de control de la congestión.

La presente tesis desarrolla un nuevo esquema para la gestión dinámica del rendimiento de *Forwarding Equivalence Classes* (FEC) en dominios *Multiprotocol Label Switching Transport Profile* (MPLS-TP) congestionados. Propone algoritmos auto-gestionados que permite a los *Label Switched Routers* (LSRs) del núcleo de la red utilizar mecanismos desde el propio

MPLS-TP para notificar eficientemente la pérdida de datos. Además, se desarrolla un método para elegir rutas más cortas para los datos retransmitidos. Todo ello hace que los datos perdidos de flujos prioritarios se recuperen más eficientemente. La propuesta recibe el nombre de *Gossip-based Local Recovery Policy* (GLRP) y se facilita como una función *Operation And Management* (OAM) para MPLS-TP. Incluye dos componentes o fases básicas: Primero, la creación del entorno de gestión, en paralelo a la señalización del *Label Switched Path* (LSP), en el Plano de Control y segundo, la notificación y recuperación de paquetes perdidos en el Plano de Reenvío. Estos componentes dan lugar a dos extensiones del protocolo de señalización *Resource Reservation Protocol with Traffic Engineering* (RSVP-TE), proporcionando la capacidad adicional que permite gobernar el comportamiento de GLRP en algunos nodos LSR ante la pérdida de paquetes de flujos prioritarios. También se analiza la funcionalidad adicional de GLRP en diferentes ámbitos, con el objetivo de facilitar la integración de GLRP en infraestructuras MPLS-TP existentes. De esta forma, se proporcionan datos analíticos y de simulación que muestran la eficiencia de la propuesta bajo diferentes condiciones. Finalmente, se identifican algunas líneas de trabajo futuro.

Abstract

The emergence of newer applications, such as data-intensive services, server virtualization, Cloud Computing or recent proposals related to Big Data, has led to a higher utilization of resources in multi-purpose Internet networks. These services imply increased bandwidth consumption, arising unexpected congestions, especially in backbones, where different types of flows coexist. It is for these dynamic variations in load that careful provisioning of the network infrastructure together with sufficient underlying capacity is necessary. However, these are not sufficient to deliver the Quality of Service (QoS) required for new applications. In operational Internet Protocol (IP) networks, it has been difficult to incorporate effective congestion control due to the limited capabilities of IP technology. In principle, Multiprotocol Label Switching Transport Profile (MPLS-TP), which is a connection-oriented label swapping technology, offers new possibilities for addressing the limitations by allowing the operator to use sophisticated traffic performance control mechanisms in next generation Connection Oriented Packet Switched networks. However, it does not include native mechanisms that address strong requirement to improve performance and efficiency of current congestion control techniques.

This thesis elaborates a novel scheme to dynamically manage the performance of prioritized Forwarding Equivalence Classes (FEC) in congested Multiprotocol Label Switching Transport Profile (MPLS-TP) domains. It proposes a self-managed algorithm that allows Label Switched Routers (LSRs) within the network to utilize mechanisms within MPLS-TP to notify packet loss earlier. Moreover, the use of shorter routes is proposed for re-transmitted data. This all results in a more efficient recovery of lost packets of prioritized flows. The proposal is known as Gossip-based Local Recovery Policy (GLRP) and is offered as an Operation and Management (OAM) function for MPLS-TP. It consists of two main components or stages: First, the establishing of the management domain in parallel with the

LSP signaling in the Control Plane and secondly notification and recovery of lost packets in the Forwarding Plane. They result in two novel extensions of the signaling protocol Resource Reservation Protocol with Traffic Engineering (RSVP-TE), which provides additional functionality in order to govern the behavior of some core LSRs when packet loss of prioritized flows arises. In addition, the extra features are discussed to improve the performance of the scheme and the ease with which the scheme can be integrated into an existing MPLS-TP infrastructure. Thus analytical and simulation data are provided in order to show the efficiency of the proposal under different conditions. Finally, topics for future research are identified.

Índice

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

Es de importancia para quien desee alcanzar una certeza en su investigación, el saber dudar a tiempo.
Aristóteles

Múltiples investigaciones se han llevado a cabo acerca del dimensionado de los recursos de red, gestión de colas o provisión de QoS en general. Estos trabajos han influido notoriamente en la calidad de los servicios proporcionados, los cuales son requeridos por tráficos prioritarios en las redes en producción sobre dominios MPLS (*Multiprotocol Label Switching*). Sin embargo, la propia imprevisibilidad o heterogeneidad del tráfico actual sigue provocando congestión en los conmutadores y routers de la red [1]. De hecho, el rápido crecimiento y penetración de nuevos tipos de servicios (*Cloud Computing*, *Big Data*, *Application & Multimedia Stores*, *Mobile VoIP*, etc.), que hacen un uso intensivo de datos o que demandan un mínimo retardo, están modificando la forma en que se consumen los recursos de red. Así, este tipo de aplicaciones suelen generar congestiones temporales (cuando aparecen nuevas actualizaciones de un sistema operativo, que es descargado masivamente, por ejemplo). Esto plantea un gran desafío al gestionar el resto del tráfico en dominios multi-servicio, sobre todo en lo que respecta al *delay* extremo a extremo (el que percibe el usuario final) o en general sobre la velocidad del resto de aplicaciones, que también pueden estar demandando fiabilidad y latencia. Por ello, tanto los proveedores de servicios de Internet, como los clientes, esperan esquemas más eficientes de ingeniería de tráfico o de gestión del uso de los recursos disponibles [2], [3].

En este contexto se está produciendo un claro movimiento hacia los servicios de paquetes que proporcionaban las redes de transporte tradicionales. Esto conlleva que la red deba evolucionar, para poder abarcar la provisión de

capacidades basadas en la conmutación de paquetes; pero esta estrategia facilita, al mismo tiempo, al proveedor de la red portadora, mantener sus inversión en infraestructura. Sin embargo, sigue siendo necesaria la implementación de nuevos esquemas de gestión más eficientes, capaces de obtener mejor fiabilidad y retardo extremo a extremo, para estos servicios de transporte de paquetes. Es por ello por lo que se está definiendo un conjunto de funciones especializadas o perfil de transporte para MPLS, conocido como *MPLS Transport Profile* [4].

En esencia, el objetivo de MPLS-TP es conseguir la convergencia con los requisitos de las redes de transporte clásicas, como escalabilidad, multiservicio, *runtime*, costes reducidos y capacidades OAM (*Operation, Administration And Maintenance*). La función PLM (*Packet Loss Measurement*) ha surgido como una de estas funciones OAM, dada la importancia de la detección y control de las pérdidas de paquetes, para los nuevos tipos de aplicaciones. De hecho, se ha convertido en un desafío clave para muchos proveedores de servicio, ya que los nuevos SLA (*Service Level Agreement*), dependerán de la capacidad que tenga la red para monitorizar estas métricas relacionadas con la pérdida de paquetes o con el retardo, evitando así la posible degradación del servicio ofrecido a los usuarios.

Si se pretende combinar los nuevos SLA con los mecanismos de control de la congestión clásicos basados en *feedback*, se debe tener en cuenta que la duración de la congestión en el dominio está directamente relacionada con el resultado del producto *latencia* * *ancho de banda*. Así, cuanto mayor sea el retardo extremo a extremo de una red, mayor será el tiempo que se tome hasta que el nodo de entrada pueda determinar que el dominio está congestionado. Además, cuanto mayor sea el ancho de banda de la red, mayor será la cantidad de datos que pondrá el emisor en la red durante el tiempo que necesite para detectar la congestión. Si además, los paquetes perdidos pertenecen a servicios con ciertos requisitos de *delay*, entonces el problema se agrava, ya que, tras detectar las pérdidas los protocolos de capas superiores, tradicionalmente lanzan retransmisiones desde el extremo origen, lo que implica retardos adicionales para dichas aplicaciones sensibles a la latencia.

En esta tesis se presenta un nuevo esquema de gestión de pérdidas basado en la arquitectura MPLS, para su funcionamiento como función OAM en dominios MPLS-TP autónomos. La propuesta diseñada extiende las capacidad de gestión de MPLS, pero al mismo tiempo puede coexistir con otros protocolos de forma transparente. El nuevo esquema ofrece:

- Mecanismos flexibles de detección de pérdidas, adecuados para funcionar dentro de un dominio MPLS multi-servicio, e incorporando características que previenen posibles inestabilidades de funcionamiento.
- Mecanismos para la notificación y retransmisión rápida de pérdidas de flujos prioritarios, basándose en una filosofía de funcionamiento local y colaborativa entre nodos y evitando al mismo tiempo señalizaciones innecesarias.
- Compatibilidad e integración en el protocolo de distribución de etiquetas RSVP-TE.

En resumen, GLRP (*Gossip-based Local Recovery Policy*) se plantea como una nueva función OAM de MPLS-TP para ofrecer un esquema de retransmisión local del tráfico perdido, en contraposición a las retransmisiones extremo a extremo clásicas. Su misión es la de proteger flujos de datos prioritarios en dominios MPLS-TP multi-servicio.

1.1 Motivación

La QoS se ha venido considerando tradicionalmente como un mecanismo para gestionar la no equidad entre diferentes flujos, los cuales deben acceder a unos recursos de red limitados. Es una forma de mejorar el funcionamiento de ciertas aplicaciones prioritarias sobre la heterogeneidad de Internet y está sustancialmente influida, entre otros parámetros, por la pérdida de datos por congestión. Por este motivo, el IETF ha estado abierto a adoptar los múltiples y novedosos modelos de servicio, políticas y mecanismos para el cumplimiento de las demandas de QoS. Además, las nuevas aplicaciones de tiempo real, tanto en redes cableadas como móviles, demandan cada vez más recursos, en entornos heterogéneos y propensos a cambios persistentes. Sin embargo, el principal desafío sigue siendo el de maximizar el uso de los recursos disponibles, mediante la implementación de mecanismos que determinen el tipo de QoS más adecuado para cada clase de servicio. Todo esto motiva nuestra investigación acerca de la gestión más eficiente de pérdidas de datos de flujos prioritarios, mediante la recuperación de paquetes en un entorno local y cooperativo.

1.2 Trabajos relacionados

Esta sección analiza el proceso de documentación, la estrategia de búsqueda para la colección de materiales de estudio básicos y las bases de conocimiento utilizadas en esta tesis, realizando finalmente una revisión de los trabajos relacionados. También se resume el trabajo existente de adaptación de protocolos clásicos de soporte de control de congestión y recuperación de pérdidas sobre los actuales canales de transmisión, sensibles a errores o con frecuentes cambios de ruta, tanto en redes cableadas como debido al *handover* de las redes móviles. De hecho, el objetivo de la revisión de la literatura existente consiste en encontrar estudios básicos que conduzcan a las posibles preguntas que se haga el investigador al intentar resolver un problema, para lo cual es importante tener en mente una adecuada estrategia de búsqueda de información. Una primera aproximación del proceso de búsqueda implica la identificación de palabras clave seleccionadas a partir del problema a resolver. Nuevas palabras clave surgen tras el análisis de los primeros artículos encontrados acerca del tema en estudio, las cuales permitirán especializar el proceso de búsqueda, así como desechar trabajos menos relacionados. Estas cadenas de búsqueda se han utilizado para coleccionar el material de estudio, accediendo para ello a un diverso conjunto de bases de datos electrónicas, incluyendo IEEE Xplore, ACM Digital Library, Springer Link, ScienceDirect (Elsevier), Engineering Village (Compendex), IET Inspec, Wiley Online Library, ISI Web of Knowledge, SCOPUS y Google Scholar, entre otros. Además, se ha mantenido un fichero índice que almacena un enlace a cada documento, para así evitar el almacenamiento de documentos repetidos y proporcionar una mejor clasificación y velocidad de acceso. De esta forma se ha indexado la documentación relacionada con nuestro trabajo, la cual se describe a continuación.

Se han venido proponiendo, desde hace décadas, múltiples modelos de *Adaptive Bandwidth Control* (ABC). Estos algoritmos ABC existentes se pueden clasificar según la técnica de control subyacente utilizada, o también en función de las métricas garantizadas de QoS. En relación a la técnica de control subyacente, los esquemas ABC pueden, a grandes rasgos, clasificarse como de ciclo cerrado o de ciclo abierto. La de ciclo cerrado, o basada en *feedback*, surge de forma natural en el contexto de pérdida de paquetes, en el que la longitud media de la cola u otros estados del sistema, son también observados para proporcionar dicho *feedback* y para ajustar el ancho de banda asignado. La propuesta de control de ciclo cerrado puede también categorizarse empleando métricas garantizadas de QoS, como la longitud media de la cola [5], la pérdida de paquetes [6] o el *delay* [7].

En su lugar, la propuesta de control de ciclo abierto implica la predicción de la ratio de tráfico entrante a partir de lo ocurrido hasta el momento en la red. La velocidad del servicio se ajusta entonces para buscar la mínima pérdida de paquetes posible, o bien el mínimo retardo en la colas. Sin embargo, conseguir un objetivo de QoS concreto es difícil debido a la no existencia de una relación directa entre la ratio de tráfico predicho y la QoS objetivo. Consecuentemente, la mayoría del trabajo existente de ABC sólo tiene por objetivo alcanzar muy pocas pérdidas, en lugar de hacer garantías cuantitativas [8]. El híbrido entre control basado en feedback y control de ciclo abierto también es posible, para así evitar los inconvenientes de ambas propuestas, como se hace en [9].

Con respecto al control de admisión, se han propuesto diversos esquemas de reparto y reserva del ancho de banda disponible, como se muestra en [10]. No obstante, la mayoría de esas propuestas proporcionan modelos analíticos para considerar el impacto del control sobre el bloqueo de conexiones para flujos de tráfico individuales y no consideran técnicas de ajuste dinámico de la capacidad, o sólo asumen condiciones de tráfico estacionario.

Por otro lado, el análisis del rendimiento de la red bajo tráfico no estacionario se ha llevado a cabo en sistemas de colas utilizando propuestas basadas en *Fluid-Flow* [11]. También los autores en [12] llevan a cabo el control de admisión con tráfico no estacionario, utilizando para ello la fórmula de *Chapman-Kolmogorov* en sistemas multi-ratio propensos a pérdidas de paquetes. En cierto modo, estos trabajos siguen sin tener en cuenta el ajuste dinámico de la capacidad disponible. Es en otros trabajos, como en [13], donde se ha estudiado el bloqueo de conexiones y el ajuste de la capacidad basándose en las condiciones de tráfico cambiantes.

Desde una perspectiva de dimensionado de la red, se ha mostrado en [14] que, en general, los requisitos de capacidad de la red son menores en condiciones de tráfico dinámico al utilizar el concepto de ruta virtual dinámica propia de las redes *Asynchronous Transfer Mode* (ATM), en comparación con la asignación estática. También en [15] se desarrolla una estrategia de asignación de capacidad dinámica, así como diversos esquemas de control de ajuste de capacidad. Estos esquemas también hacen uso de la propuesta *Fluid-Flow* para analizar el tráfico dinámico en sistemas propensos a pérdidas, basándose en el bloqueo y en el uso de la red como medio para calcular cuándo y cuánto ajuste de capacidad debería hacerse. Si bien, el inconveniente de estos esquemas es la asunción de un conocimiento previo del tráfico, para que los parámetros de ajuste puedan ser adecuadamente calculados y suministrados.

Al mismo tiempo, algunas variantes de *Transport Control Protocol* (TCP) proponen inferir el exceso de carga en la red a partir del *feedback* recibido de la propia red [16] o basarse en el RTT [17], para adaptar TCP a las redes de alta velocidad. En el primer caso, se requiere el soporte de este *feedback* por parte de todos los routers intermedios por los que circula el tráfico, lo cual difícilmente puede ser logrado debido a la propia extensión y heterogeneidad de Internet. Para el segundo caso, conocido como variantes TCP basadas en *delay*, las propuestas puede que no trabajen bien sobre redes en las que sean comunes los cambios de rutas y reordenaciones de paquetes, lo cual puede hacer que el RTT varíe en función de factores diferentes al de la carga de la red. Al mismo tiempo, las implementaciones clásicas de TCP basadas en pérdidas más utilizadas en la actualidad no suelen hacer un reparto equitativo del ancho de banda disponible [18].

Las implementaciones TCP-DCR en [19], AVG, DEL, EWMA e INC en [20], RR-TTP en [21] y TCP-PR en [22] proponen abandonar el método clásico de esperar tres ACKs duplicados como señal de pérdida de paquetes por congestión. En su lugar proponen retrasar proactivamente la retransmisión de un paquete desde el origen hasta que un temporizador asociado expire o cuando el número de ACKs duplicados recibidos alcance un valor umbral, que cambia dinámicamente. A priori esta estrategia parece más fiable a la hora de detectar pérdidas en canales en los que suele ser necesaria la reordenación de paquetes. Sin embargo, estas variantes siguen teniendo en cuenta el número de ACKs duplicados recibidos como indicio de congestión y así activar la respuesta, por lo que siguen sin poder diferenciar entre pérdidas por congestión y simples reordenamientos de paquetes debidos a cambios de ruta u otros motivos.

DSACK TCP [23], TCP-DOOR [24] y TCP-Eifel [25] se diseñan con la premisa de que el triple ACK duplicado puede que no sea una indicación fiable de pérdidas por congestión. Sus aproximaciones difieren, sin embargo, en que intentan detectar retransmisiones innecesarias tras la llegada del triple ACK duplicado. En cuanto se produce una detección, *cwnd* se recupera inmediatamente, o bien se inicia el proceso clásico de arranque lento. TCP-DOOR, además, desactiva el control de congestión durante un intervalo de tiempo, ya que esta propuesta asume que la llegada de confirmaciones de paquetes fuera de orden viene provocada a menudo por cambios en la ruta. El inconveniente es que estas variantes, generalmente, no son capaces de recuperar una *cwnd* que ha sido innecesariamente reducida por pérdidas de paquete cuando los motivos han sido diferentes a la congestión. Además, como se muestra en [26], TCP-DOOR tiende a desactivar excesivamente las acciones de

control de congestión, en situaciones de reordenación de paquetes persistente, lo que provoca un substancial deterioro del rendimiento.

JTCP [27], TCP-Veno [28] y TCP-Westwood [29] se centran en la diferenciación entre las pérdidas por congestión y las debidas a otros motivos, empleando una estimación de la carga de red en el momento de la llegada del triple ACK duplicado. Por tanto, esta señal todavía es considerada por estas propuestas como un indicio creíble de pérdida de paquetes. No obstante, esta señal por sí sola no dicta la activación de la respuesta a la congestión, sino que se combina con la estimación de la carga de la red, para tomar la decisión de cuánto se debería reducir *cwnd*. Todavía la inherente asunción de un canal formado por paquetes ordenados, obviamente limita la aplicabilidad de estas variantes sobre, por ejemplo, redes móviles o en otras muchas redes actuales, en las que la reordenación de paquetes suele ser necesaria. Así mismo, en redes donde existe reordenación persistente de paquetes, la búsqueda de la convergencia entre *cwnd* y el ancho de banda disponible puede verse significativamente afectada, debido a estas pequeñas pero frecuentes reducciones de *cwnd*.

TCP-Probing [30] aplica una aproximación diferente para diferenciar entre pérdidas por congestión y las debidas a otros motivos. Después de que *cwnd* se haya reducido a la mitad en respuesta a la pérdida de paquetes inferida a partir de un triple ACK duplicado, algunos paquetes de datos de prueba se inyectan en la red. Si el RTT de los dos primeros paquetes confirmados es más pequeño que el mejor RTT medido durante la sesión TCP, entonces se considera que la recepción del triple ACK duplicado se ha debido a motivos ajenos a la congestión. De nuevo TCP-Probing disparará frecuentes retransmisiones innecesarias en caso de redes con reordenación de paquetes persistente. Además, el mejor RTT almacenado puede fallar si se están produciendo cambios en las rutas de la red. Por ejemplo, cuando tiene lugar un cambio de ruta o un *handoff* en una red móvil durante una sesión TCP y se incrementa el mínimo RTT alcanzable, el mejor RTT almacenado no se actualizará. Consecuentemente, esto puede provocar que los paquetes de prueba no se puedan confirmar dentro del mejor RTT, incluso con baja carga de tráfico en la red.

Existen soluciones unificadas que buscan lo mejor de varias propuestas, pero requieren información o modificaciones de la pila de protocolos de varias capas, más allá de la de transporte. Por ejemplo, la propuesta ATCP [31] introduce la capa ATCP entre los niveles TCP e IP. Esta nueva capa conmuta TCP entre varios estados predefinidos, de acuerdo con la condición de la red, intentando así

evitar respuestas a la congestión y retransmisiones innecesarias de paquetes desde el origen.

Ramdom Early Detection, propuesto por primera vez en 1993 [32], está recibiendo múltiples propuestas en la actualidad, buscando la inherente ventaja de AQM a la hora de ayudar a la red a operar en la región óptima de *throughput* elevado y *delay* reducido. Sin embargo, los nuevos ajustes de los parámetros de funcionamiento implicados y las nuevas variantes de RED adoptan de nuevo la aproximación del triple ACK duplicado para inferir errores, ya que ésta es una estrategia más sencilla y con menos dificultades que la dinámica de funcionamiento TCP/AQM. Por ejemplo, los modelos presentados en [33] proporcionan una interesante base para el diseño y análisis sistemático de AQM. La aproximación basada en optimización [34] interpreta TCP/AQM como un algoritmo distribuido para resolver un problema de maximización de uso, sujeto a restricciones de capacidad. El foco se centra en las soluciones óptimas alcanzadas en el equilibrio, pero las respuestas transitorias de AQM se suelen desechar. Estas aproximaciones teóricas consideran el control de la congestión como un sistema de control no lineal. El sistema primero se convierte en lineal alrededor de su punto de equilibrio, con el objetivo de analizar la dinámica de TCP/AQM alrededor del equilibrio. Esto permite el ajuste de los parámetros de RED y el diseño de nuevos algoritmos AQM basados en el análisis en el dominio de la frecuencia [35], lo cual es útil para la mejora de la respuesta transitoria de AQM y para garantizar la estabilidad del sistema lineal, lo cual se revisa en [36].

Conceptos como la *estabilidad global* o la *región de atracción* del control de la congestión en Internet también se han estudiado en el contexto de sistemas no lineales [37]. En pocas palabras, estas propuestas están relacionadas con el hecho de que el sistema debe converger al equilibrio comenzando en un estado inicial factible no necesariamente cercano al equilibrio [38]. En cualquier caso, la gestión AQM todavía se basa en una ratio estática, lo que determina que el borrado proactivo de paquetes de AQM esté en función de la velocidad de llegada de nuevos paquetes en un instante dado.

Otras propuestas ampliamente utilizadas en Internet consideran variantes AQM que determinan la ratio de borrado de paquetes a partir del tamaño de la cola, como en [39], en el que los autores adaptan AQM a redes móviles, mejorando virtualmente la equidad entre flujos en redes ad-hoc. Sin embargo, generalmente AQM no permite estabilizar el tamaño de la cola, ni maximizar el *throughput* en el núcleo de la red. Además, como ya se ha comentado, los nodos proporcionan *feedback* a los emisores mediante el descarte proactivo de paquetes,

lo que puede provocar que los protocolos de capas superiores respondan a estos pequeños incrementos en la ratio de pérdida con grandes reducciones en la ratio de envío, debido a la reducción de *cwnd*.

En [40] se describe una evaluación del rendimiento de la red ante pérdidas, así como un análisis comparativo del retardo entre redes IPv6 e IPv4. En sus trabajos también presentan y discuten la metodología de medida empleada, llegando a la conclusión de que las rutas configuradas mediante IPv6 presentan mayor retardo y pérdida de paquetes que las de IPv4. En este sentido, también en [41] se analizan los problemas de implementación de servicios con QoS en IPv6, así como su rendimiento, ventajas de las tecnologías de transición actuales y problemas encontrados. También proporcionan una comparativa de rendimiento entre tres diferentes mecanismos de transición: encapsulado *IPv6-in-IPv4*, encapsulado *6PE* (IPv6 sobre una red MPLS-IPv4) y tecnologías de red de área local de banda ancha.

Así mismo, en [42] se lleva a cabo una comparativa de la garantía de QoS que ofrecen los tres esquemas clásicos más comunes: Servicios Integrados (*IntServ*), Servicios Diferenciados (*DiffServ*) y QoS sobre IPv6. En los resultados obtenidos de sus pruebas se muestra que la gestión QoS sobre IPv6 consigue mejores resultados que *IntServ* o *DiffServ*. También, el artículo [43] evalúa los mecanismos de QoS sobre dominios IPv6, con el objetivo de proporcionar servicios de tráfico agregado de tiempo real, con mínimo *delay*, *jitter* y pérdida de paquetes. En sus experimentos, como ejemplo de tráfico de tiempo real, han considerado el proyecto *OpenH323*, una implementación de fuentes abiertas de *H.323*. En general, sus resultados muestran que el nivel de QoS que experimenta el tráfico de tiempo real es adecuado y da a las aplicaciones de tiempo real la habilidad de operar con estándares de alta calidad, sin consecuencias significativas sobre el tráfico principal.

En [44] se presenta un servicio de QoS implementado sobre una red IPv6 nativa a gran escala, concluyendo que los mecanismos de QoS clásicos de clasificación, priorización y vigilancia del cumplimiento de requisitos funcionan adecuadamente. En el mismo contexto de redes de área extensa, los autores en [45] discuten acerca de las pruebas y de los desafíos técnicos que se deben afrontar durante el despliegue de mecanismos de QoS para IPv6 en el núcleo de las redes troncales. En particular centran su trabajo en la red en producción *GRNET* (*Greek Research and Education Network*) y en la red piloto europea *6NET,* utilizando plataformas tanto hardware como software, como se detalla en los proyectos [46] al [50]. Demuestran que se pueden transmitir servicios

avanzados de transporte mediante tráfico IPv6 con éxito y con diferentes niveles de garantía de funcionamiento. Uno de los principales hallazgos en sus pruebas es que los mecanismos de QoS para IPv6 son soportados eficientemente por las interfaces de routers actuales, incluso a velocidades de varios Gigabits por segundo.

Por añadidura, en [51] se describe la implementación de un *testbed* en el que se interconectan tres dominios *DiffServ* mediante el empleo de túneles estáticos *IPv6-in-IPv4*. Desarrollan un estudio acerca de los problemas de rendimiento, analizando el *throughput*, la pérdida de paquetes y el *delay* en servicios relacionados con la aviación, como CPDLC (*Controller to Pilot Data Link Communication)*, mediante el empleo de *DiffServ* sobre una red troncal basada en IPv6. Los resultados que obtienen confirman que la combinación *DiffServ* e IPv6 está lo suficientemente madura como para proporcionar QoS estable y fiable en aplicaciones relacionadas con la aviación. Utilizando también una estrategia basada en *testbeds*, en [52] se lleva a cabo la implementación de diversos mecanismos para proporcionar QoS empleando dispositivos de red reales.

De igual forma, en [53] se detalla el empleo de procesadores de red reales para estudiar cómo implantar *DiffServ* para la evaluación de su capacidad de priorización de flujos. Para ello desarrollan y prueban un método de priorización basado en WRED, con el objetivo de proporcionar QoS basada en reglas, es decir, en función de la clase de los paquetes y del tráfico existente en la red.

En [54] los autores evalúan el rendimiento de transmisiones TCP y UDP bajo diferentes entornos de red. En particular se centran en el análisis de una red nativa IPv4, otra nativa IPv6 y una red IPv6 encapsulada en IPv4, utilizando para ello *MPLS-Linux*. Con respecto a la construcción del entorno emplean máquinas virtuales funcionando sobre Linux y como herramienta de medida utilizan *Iperf*. Han demostrado que el rendimiento de las transmisiones TCP tanto en IPv4 como en IPv6 es casi el mismo, mientras que el rendimiento de la transmisión TCP en el caso de IPv6 encapsulado en IPv4 es menor, comparado con los rendimientos de IPv4 e IPv6 nativos. Además, las gráficas de rendimiento de la transmisión UDP en los tres entornos están próximas entre sí.

1.3 Definición del problema

A partir de los mecanismos de control de pérdidas analizados, se deduce que no existe una solución unificada de señalización de la pérdida de paquetes, para los múltiples y heterogéneos servicios prioritarios que están surgiendo en Internet. Como se ha comentado, todas las propuestas centran su esfuerzo en mejorar el proceso de detección de pérdidas llevado a cabo por los nodos del borde. Sin embargo, esta estrategia presenta una latencia de recuperaciones bastante pobre, ya que, al elevado intervalo de tiempo requerido para detectar las pérdidas, hay que sumar que se siguen llevando a cabo retransmisiones extremo a extremo de los paquetes perdidos. En este contexto, GLRP lleva la detección de pérdidas a capas inferiores, en particular al Plano MPLS, con el objetivo de obtener detecciones de pérdidas de datos de flujos prioritarios más rápidas. Para ello, GLRP establecerá un entorno colaborativo de nodos intermedios, los cuales se notificarán entre ellos acerca de las pérdidas producidas. Además, GLRP dará lugar a retransmisiones locales entre los nodos de dicho entorno, con el objetivo de mejorar también la latencia de las retransmisiones de paquetes perdidos.

1.4 Objetivos

En secciones anteriores se ha destacado la importancia del desarrollo de un esquema de control de pérdidas de paquetes por congestión, adecuado para MPLS. Así, el principal objetivo de la investigación de esta tesis ha sido la detección eficiente de pérdidas de datos de flujos prioritarios, así como la reducción de la latencia de las retransmisiones de dichos datos, ya que se lleva a cabo en un entorno local, en lugar de hacerlo extremo a extremo. Estos objetivos, a su vez, se pueden desglosar en otros más concretos:

- Desarrollar un esquema de gestión de pérdidas de paquetes, para detectar y aliviar la congestión en redes de área extensa multi-servicio y de forma transparente para otros protocolos.
- Seleccionar un método eficiente de colaboración entre nodos para la señalización de la información en situaciones de congestión y de esta manera minimizar sus efectos.
- Asegurar que el esquema propuesto interacciona con los estándares del IETF desarrollados para MPLS o cualquiera de sus variantes. Esto proporciona una forma de migración sencilla para los operadores de red.

Para cumplir los objetivos anteriores, la fase inicial de la investigación se ha concentrado en la necesidad de establecer un mecanismo de señalización de la información de congestión entre nodos MPLS. Para ello se emplea el propio protocolo de señalización de rutas, con el objetivo de obtener un método eficiente y transparente para otros protocolos. En cualquier caso, también se presenta una descripción general de MPLS y sus variantes, así como de sus principales desafíos en las redes de nueva generación. En particular, se destaca la importancia de clasificar los servicios en función de su nivel de prioridad, lo cual condicionará el tratamiento que se dé a los paquetes de datos. De esta forma, se pretende mejorar la percepción que pueden llegar a tener los clientes de servicios garantizados, que se han identificado como prioritarios, lo que ha conducido al desarrollo de nuestro esquema de gestión de pérdidas.

1.5 Resumen de contribuciones

Comenzando con la premisa de que las troncales de nueva generación serán, por naturaleza, de tipo multi-servicio, en este trabajo se ha acometido un recorrido por los principales mecanismos de control de QoS que lo soportan. Se han analizado trabajos que proponen nuevos mecanismos que optimizan el reparto de recursos (*capacity allocation*), otros relacionados con el control eficiente de la congestión y de las pérdidas, de ingeniería de tráfico o, en general, de control de la QoS. En especial se ha examinado MPLS, dado su fuerte respaldo comercial y su capacidad de penetración en los nuevos mecanismos de QoS de la futura Internet. Sin embargo, hemos determinado que, aunque es un protocolo relativamente flexible, en su forma actual no proporciona medios de implementación de control de pérdidas de rápida actuación.

Dado este déficit, esta tesis centra su foco en el diseño de un esquema de control de pérdidas de paquetes que se construye sobre los conceptos básicos de MPLS y sus nuevas variantes, pero que proporciona medios para resolver rápidamente los efectos adversos de dichas pérdidas. Este esquema va más lejos del tradicional sobre-aprovisionamiento y es más viable para proporcionar una ventaja significativa al mercado de los fabricantes de equipos de red que implementen el esquema o sus derivados.

El esquema propuesto, llamado GLRP, opera sobre dominios MPLS-TP multi-servicio, protegiendo flujos prioritarios ante pérdidas de paquetes por congestión o debidas a cambios persistentes de ruta. Su viabilidad ha sido analizada siguiendo, tanto un modelo de optimización, como otro probabilístico. También se ha implementado un modelo de simulación para asegurar su correcto

funcionamiento y su rendimiento satisfactorio. Si bien, también fue necesario abordar una serie de refinamientos sobre el esquema durante el curso de la investigación. Estos incluyen:

- La distinción entre flujos MPLS etiquetados y no etiquetados que dio sentido a la variante CL-GLRP (*Connectionless GLRP*), como servicio especializado para flujos de paquetes no etiquetados.
- La adopción de RSVP-TE como protocolo para la señalización de la información de GLRP, evitando introducir *overhead* adicional en el Plano de Datos a la hora de crear el entorno operacional de nuestra propuesta, ya que pasó a crearse en paralelo a la reserva de recursos y configuración del LSP.
- El análisis probabilístico fue añadido para permitir a los nodos con capacidad GLRP decidir si existe probabilidad suficiente de recuperación de una pérdida antes de iniciar el proceso de señalización, evitando así notificaciones innecesarias.
- La introducción del concepto *diámetro máximo* mejoró la estabilidad del esquema, ya que es un umbral que limita la generación de mensajes de notificación de pérdidas, las cuales también se pueden ahora agregar dentro de mensajes RSVP-TE, evitando los efectos de avalancha a lo largo de la ruta.

Finalmente se evaluó la eficacia de la propuesta, la cual fue sometida al proceso de revisión en múltiples congresos y revistas, tanto de ámbito nacional como internacional. GLRP ha sido considerada como un original y eficaz mecanismo de gestión de pérdidas, ubicado en niveles inferiores de la torre de protocolos. Esto origina que la recuperación de pérdidas se lleve a cabo de una forma más eficiente que mediante el empleo de sistemas de retransmisión extremo a extremo, propios de protocolos de capas superiores. GLRP, como esquema de control de pérdidas de rápida actuación, ha sido valorada potencialmente útil en la industria, para la gestión de pérdidas en dominios MPLS multi-servicio, en los que confluyan flujos de datos para los que el retardo es un parámetro clave.

1.6 Estructura de la tesis

El esquema de la tesis se organiza de la siguiente manera: En este capítulo, la introducción, se ha discutido el trabajo de fondo, la motivación, los objetivos, las principales contribuciones y una revisión de trabajos relacionados. En el Capítulo 2 se detallan los aspectos más relevantes de MPLS, MPLS-TE y MPLS-TP, incidiendo sobre sus capacidades de ingeniería de tráfico, control de la congestión y QoS. También se elabora brevemente en este capítulo una discusión sobre cómo un algoritmo basado en *Constrained Shortest Path First* (CSPF) puede colaborar a la hora de encontrar la mejor ruta entre dos nodos, en función de los requisitos de QoS del nuevo flujo de datos. Finalmente se hace un primer esbozo de los potenciales beneficios que puede aportar GLRP en un dominio MPLS-TP. En el Capítulo 3 se define nuestra propuesta GLRP, explicando en detalle su integración con el protocolo de señalización RSVP-TE para dar servicio a un dominio MPLS. También se muestra la arquitectura sencilla del nodo con capacidad GLRP. Además, se analizan otras capacidades de GLRP, como la ofrecida para servicios no orientados a conexión, la recuperación local de ráfagas de paquetes, la reordenación de paquetes o el funcionamiento de GLRP en dominios punto-multipunto. En el Capítulo 4 se hace un análisis matemático sobre la viabilidad de GLRP, así como un estudio probabilístico de la misma. En el Capítulo 5 se lleva a cabo una discusión sobre los métodos de simulación utilizados para generar los modelos de red, destacando *OpenSimMPLS*v2 y Network Simulator. Se proporciona, además, la necesaria información experimental para permitir una completa y comprensiva evaluación del esquema propuesto. También se muestran y analizan los resultados de simulación obtenidos. Finalmente el Capítulo 6 concluye el trabajo de investigación, analizando también algunos posibles trabajos futuros.

1.7 Referencias del capítulo

[1] Euiyul Ko, Donghyeok An, Ikjun Yeom and Hyunsoo Yoon, “Congestion control for sudden bandwidth changes in TCP,” International Journal of Communication Systems, vol 25, nº 12. Pag. 1550-1567. 2012. [Online]: http://dx.doi.org/10.1002/dac.1322 (último acceso abril 2014).

[2] G. Papastergiou, C. Georgiou, L. Mamatas and V. Tsaoussidis, “A delay-oriented prioritization policy based on non-congestive queuing,” International Journal of Comm. Systems, vol 24, nº 8. Pág. 1065-1086. 2011. [Online]: http://dx.doi.org/10.1002/dac.1215 (último acceso abril 2014).

[3] Y. Zhao, D. Han, J. Zhang, J. Xing, “QoS sensitive routing in DiffServ MPLS-TP networks,” International Conference on Computer Application and System Modeling, ICCASM '10, vol. 1. Pág. 726–730. 2010. [Online]: http://dx.doi.org/10.1109/ICCASM.2010.5620187 (último acceso abril 2014).

[4] M. Bocci et al., “A Framework for MPLS in Transport Networks,” IETF Request for Comments 5921, Standards Track. 2010.

[5] A. Pitsillides, P. Ioannou and L. Rossides, "Congestion Control for Differentiated Services using Non-linear Control Theory," Sixth IEEE Symposium on Computers and Comm. Pág. 726-733. 2001. [Online]: http://dx.doi.org/10.1109/ISCC.2001.935456 (último acceso abril 2014).

[6] P. Siripongwutikorn, S. Banerjee and D. Tipper, "Adaptive Bandwidth Control for Efficient Aggregate QoS Provisioning," IEEE Global Telecomm. Conference, GLOBECOM '02, vol 3. Pág. 2435 - 2439. 2002. [Online]: http://dx.doi.org/10.1109/GLOCOM.2002.1189068 (último acceso abril 2014).

[7] N. Christin, J. Liebeherr and T. F. Abdelzaher, "A Quantitative Assured Forwarding Service," IEEE INFOCOM '02, vol 2. Pág. 864-873. 2002. [Online]: http://dx.doi.org/10.1109/INFCOM.2002.1019333 (último acceso abril 2014).

[8] P. Siripongwutikorn, S. Banerjee and D. Tipper, "A Survey of Adaptive Bandwidth Control Algorithms," Comm. Surveys & Tutorials, vol. 5, nº. 1. Pág. 14-26. 2003. [Online]: http://dx.doi.org/10.1109/COMST.2003.5342227 (último acceso abril 2014).

[9] Z. Sahinoglu and S. Tekinay, "A Novel Approach Bandwidth Allocation: Wavelet-Decomposed Signal Energy Approach," IEEE Global Telecommunications Conference, GLOBECOM '01, vol 4. pp 2253-2257. 2001. [Online]: http://dx.doi.org/10.1109/GLOCOM.2001.966180 (último acceso abril 2014).

[10] D. Medhi, A. van de Lievoort and C. S. Reece, "Performance Analysis of a Digital Link with Heterogeneous Multislot Traffic," IEEE Transactions on Communications, vol. 43, nº 234. Pág. 968-976. 1995. [Online]: http://dx.doi.org/10.1109/26.380129 (último acceso abril 2014).

[11] W. Wang, D. Tipper and S. Banerjee, "A Simple Approximation for Modeling Nonstationary Queues," IEEE INFOCOM '96, vol 1. Pág. 255-262. 1996. [Online]: http://dx.doi.org/10.1109/INFCOM.1996.497901 (último acceso abril 2014).

[12] Y. Qian, D.Tipper and D. Medhi, "An Analysis of Access Control Schemes for Multirate Loss Networks Under Nonstationary Conditions," IEEE INFOCOM

vol 2. Pág.730-737, 1996. [Online]: http://dl.acm.org/citation.cfm?id=1895920 (último acceso abril 2014).

[13] S. Shioda, H. Toyoizumi, H. Yokoi, T. Tsuchiya and H. Saito, "Self-sizing Network: A New Concept Based on Autonomous Network VP Bandwidth Adjustment," Proceedings of 15th International Teletraffic Congress, ITC15. Pág. 997-1006, 1997.

[14] D. Medhi, "Multi-hour Network Design for Dynamically Reconfigurable Wide-Area ATM Networks," IEEE/ACM Transactions on Networking, vol. 3, nº 6. Pág. 808-819. 1995. [Online]: http://dx.doi.org/10.1109/90.477726 (último acceso abril 2014).

[15] B. Groskinsky, D.Medhi and D. Tipper, "An Investigation Of Adaptive Capacity Control Schemes In A Dynamic Traffic Environment", IEICE Transactions on Comm, vol. E00-A, nº 13. Pág. 263-274. 2001. [Online]: http://sce.umkc.edu/~dmedhi/papers/old/gmt_ieice_2001.pdf (último acceso abril 2014).

[16] Y. Xia, L. Subramanian, I. Stoica and S. Kalyannaramana, "One More Bit is Enough,". IEEE/ACM Trans. on Networking, Vol. 16, nº. 6, 2008. [Online]: http://dx.doi.org/10.1109/TNET.2007.912037 (último acceso abril 2014).

[17] D.Wei, C. Jin, S. Low and S. Hegde, "FAST TCP: Motivation, Architecture, Algorithms, Performance," IEEE/ACM Trans. on Networking, Vol. 14, nº. 6. Pág. 1246-1259. 2006. [Online]: http://dx.doi.org/10.1109/TNET.2006.886335 (último acceso abril 2014).

[18] A. Tang, J. Wang, S.H. Low and M. Chiang, "Equilibrium of Heterogeneous Congestion Control: Existence and Uniqueness," IEEE/ACM Transactions on Networking, Vol. 15, nº. 4. Pág. 1422-1435. 2007. [Online]: http://dx.doi.org/10.1109/TNET.2007.893885 (último acceso abril 2014).

[19] S. Bhandarkar and A.L.N. Reddy, "TCP-DCR: Making TCP Robust to Non-Congestion Events," Lecture Notes in Computer Science, Vol. 3042. Pág. 712-724. 2004. [Online]: http://dx.doi.org/10.1007/978-3-540-24693-0_59 (último acceso abril 2014).

[20] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering". ACM SIGCOMM Computer Communication Review, Vol. 32, no 1. Pág. 20-30. 2002. [Online]: http://dl.acm.org/citation.cfm?id=510728 (último acceso abril 2014).

[21] M. Zhang, B. Karp, S. Floyd and L. Peterson, "RR-TCP: A Reordering-Robust TCP with DSACK," 11th IEEE Int. Conference on Network Protocols. Pág. 95-

106. 2003. [Online]: http://dx.doi.org/10.1109/ICNP.2003.1249760 (último acceso abril 2014).

[22] S. Bohacek, J.P. Hespanha, J. Lee, C. Lim and K. Obraczka, "A New TCP for Persistent Packet Reordering," IEEE/ACM Trans. on Networking," vol. 14, nº. 2. Pág. 369-382. 2006. [Online]: http://dx.doi.org/10.1109/ICDCS.2003.1203469 (último acceso abril 2014).

[23] S. Floyd, J. Mahdavi, M. Mathis and M. Podolsky, "An Extension to the Selective Acknowledgment (SACK) Option for TCP," IETF Request for Comments 2883, Standards Track. 2000.

[24] F. Wang and Y. Zhang. "Improving TCP Performance over Mobile Ad-Hoc Networks with Out-Of-Order Detection and Response". 3rd ACM international symposium on Mobile ad hoc networking & computing MOBIHOC '02. Pág. 217-225. 2002. [Online]: http://dl.acm.org/citation.cfm?id=513827 (último acceso abril 2014).

[25] R. Ludwig and R.H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," ACM SIGCOMM Computer Comm. Review, Vol. 30, nº 1. Pág. 30-36. 2000. [Online]: http://dl.acm.org/citation.cfm?id=505692 (último acceso abril 2014).

[26] Chengdi Lai, Ka-Cheong Leung and V. Li, "Does it hurt when others prosper?: Exploring the impact of heterogeneous reordering robustness of TCP," IEEE INFOCOM 2013. [Online]: http://dx.doi.org/10.1109/INFCOM.2013.6567107 (último acceso abril 2014).

[27] E. H.-K. Wu and M.-Z. Chen, "JTCP: Jitter-Based TCP for Heterogeneous Wireless Network," IEEE Journal on Selected Areas in Comm., Vol. 22, nº. 4. Pág. 757-766. 2004. [Online]: http://dx.doi.org/10.1109/JSAC.2004.825999 (último acceso abril 2014).

[28] C.P. Fu and S.C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," IEEE Journal on Sel. Areas in Comm. Vol. 21, nº. 2. Pág. 216-228. 2003. [Online]: http://dx.doi.org/10.1109/JSAC.2002.807336 (último acceso abril 2014).

[29] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," Journal on Wireless Networks, Vol. 8, nº. 5. Pág. 467-479. 2002. [Online]: http://dl.acm.org/citation.cfm?id=582460 (último acceso abril 2014).

[30] A. Lahanas and V. Tsaoussidis, "Improving TCP Performance over Networks with Wireless Components using 'Probing Devices'," International Journal of

Communication Systems, Vol. 15, nº. 6. Pág. 495-511. 2002. [Online]: http://dx.doi.org/10.1002/dac.548 (último acceso abril 2014).

[31] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," IEEE Journal on Selected Areas in Communication, Vol. 19, nº. 7. Pág. 1300-1315. 2001. [Online]: http://dx.doi.org/10.1109/49.932698 (último acceso abril 2014).

[32] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE Transactions on Networking, Vol. 1, nº. 4. 1993. [Online]: http://dl.acm.org/citation.cfm?id=169935 (último acceso abril 2014).

[33] S.H. Low, F. Paganini and J.C. Doyle, "Internet Congestion Control," IEEE Control Systems Magazine, Vol. 22, nº. 1. Pág. 28-43. 2002. [Online]: http://dx.doi.org/10.1109/37.980245 (último acceso abril 2014).

[34] S.H. Low, "A Duality Model of TCP and Queue Management Algorithms," IEEE/ACM Trans. on Networking, Vol. 11, nº. 4. Pág. 525-536. 2003. [Online]: http://dx.doi.org/10.1109/TNET.2003.815297 (último acceso abril 2014).

[35] H. Han, C.V. Hollot, Y. Chait and V. Misra, "TCP Networks Stabilized by Buffer-Based AQMs," IEEE INFOCOM '04, vol 2. Pág. 964-974. 2004. [Online]: http://dx.doi.org/10.1109/INFCOM.2004.1356983 (último acceso abril 2014).

[36] R. Srikant, "The Mathematics of Internet Congestion Control," Ed. Springer-Verlag. 2004. ISBN 978-0-8176-8216-3.

[37] L. Ying, G.E. Dullerud and R. Srikant, "Global Stability of Internet Congestion Controllers with Heterogeneous Delays," IEEE/ACM Transactions on Networking, Vol. 14, nº. 3. Pág. 579-590. 2006. [Online]: http://dx.doi.org/10.1109/TNET.2006.876164 (último acceso abril 2014).

[38] H.K. Khalil, "Nonlinear Systems," 3rd. Edition. Ed Prentice-Hall. 2002. ISBN 0-13-067389-7.

[39] K. Xu, M. Gerla, L. Qi and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED," 9th ACM Annual Int. Conf. on Mobile Computing and Networking, MOBICOM '03. pp 16-28. 2003. [Online]: http://dl.acm.org/citation.cfm?id=938988 (último acceso abril 2014).

[40] J. Hanumanthappa and D.H. Manjaiah, "IPv6 over IPv4 QoS metrics in 4G Networks: Delay, Jitter, Packet Loss Performance, Throughput and Tunnel Discovery," International Conference on Information Science and Applications, ICISA'10. 2010.

[41] Y. Adam, B. Fillinger, I. Astic, A. Lahmadi and P. Brigant, "Deployment and test of IPv6 services in the VTHD network," IEEE Comm. Magazine, vol. 42.

Pág. 98–104. 2004. [Online]: http://dx.doi.org/10.1109/MCOM.2004.1262168 (último acceso abril 2014).

[42] E. Fgee, J. Kenney, W. Phillips, W. Robertson and S. Sivakumar, "Comparison of QoS performance between IPv6 QoS management model and IntServ and DiffServ QoS models," 3rd Annual IEEE Communication Networks and Services Research Conference. 2005. Pág. 287–292. [Online]: http://dx.doi.org/10.1109/CNSR.2005.28 (último acceso abril 2014).

[43] C. Bouras, A. Gkamas, D. Primpas, and K. Stamos, "Performance Evaluation of the Impact of QoS Mechanisms in an IPv6 Network for IPv6-Capable Real-Time Applications," Journal of Network and Systems Management, vol. 12, no. 4. Pág. 463–483. 2004. [Online]: http://dx.doi.org/10.1007/s10922-004-0672-5 (último acceso abril 2014).

[44] C. Bouras, A. Gkamas, D. Primpas and K. Stamos, "IPv6 deployment: Real time applications and QoS aspects," Computer comm., vol. 29, nº. 9. Pág. 1393–1401. 2006. [Online]: http://dx.doi.org/10.1016/j.comcom.2005.08.014 (último acceso abril 2014).

[45] A. Liakopoulos, D. Kalogeras, V. Maglaris, D. Primpas, and C. Bouras, "QoS experiences in native IPv6 networks," International Journal of Network Management, vol. 19, nº. 2. Pág. 119–137. 2009. [Online]: http://dx.doi.org/10.1002/nem.695 (último acceso abril 2014).

[46] Euro6IX, "IPv6 Deployment in Europe." [Online]: http://www.euro6ix.org/main/ (último acceso abril 2014).

[47] 6INIT. Promoting the introduction of IPv6 multimedia and security. [Online]: http://www.6init.org/ (último acceso abril 2014).

[48] 6POWER. The deployment of IPv6 in Europe. [Online]: http://www.6power.org/ (último acceso abril 2014).

[49] 6QM. IPv6 QoS Measurement. [Online]: http://www.6qm.org/ (último acceso abril 2014).

[50] SATIP6. Satellite broadband multimedia system for IPv6. [Online]: www.ist-world.org/ProjectDetails.aspx?ProjectId=659243fd480d42308d1553fa7519cb4b (último acceso abril 2014).

[51] V. Srivastava, C. Wargo and S. Lai, "Aviation application over IPv6: performance issues," IEEE Aerospace Conf, vol. 3. 2004. [Online]: http://dx.doi.org/10.1109/AERO.2004.1367941 (último acceso abril 2014).

[52] C. Bouras, A. Gkamas, D. Primpas and K. Stamos, "Quality of Service aspects in an IPv6 domain," International Symposium on Performance Evaluation of

Computer and Telecom. Systems (SPECTS '04). 2004. Pág. 238–245. [Online]: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.8722 (último acceso abril 2014).

[53] K. Saleem, N. Fisal and M. Zabidi, "QoS provisioning for real time services on DiffServ aware IPv6 optical network using IXP-2400 Intel Network," 5th IFIP IEEE Int. Conference on Wireless and Optical Comm. Networks, WOCN'08. 2008. Pág. 1–5. [Online]: http://dx.doi.org/10.1109/WOCN.2008.4542489 (último acceso abril 2014).

[54] R. Yunos, N. Noor and S. Ahmad, "Performance evaluation between IPv4 and IPv6 on MPLS Linux platform," IEEE International Conference on Information Retrieval & Knowledge Management, CAMP '10. 2010. Pág. 204–208. [Online]: http://dx.doi.org/10.1109/INFRKM.2010.5466916 (último acceso abril 2014).

Capítulo 2. MultiProtocol Label Switching

Cualquier tecnología suficientemente avanzada es indistinguible de la magia.
Arthur Clarke

En el anterior capítulo, se ha discutido el trabajo de fondo, la motivación, los objetivos, las principales contribuciones y una revisión de trabajos relacionados. En este capítulo se detallarán los aspectos más relevantes de MPLS, MPLS-TE y MPLS-TP, incidiendo sobre sus capacidades de ingeniería de tráfico, control de la congestión y provisión de QoS.

2.1 Fundamentos de MPLS

El protocolo MPLS ha suscitado múltiples esfuerzos y propuestas, muchas de las cuales han tenido un gran impacto sobre las redes IP. Las diferentes técnicas MPLS se han ido implementando tanto en las redes del proveedor de servicio como en las redes de transporte, lo que ha implicado el remodelado continuo de las arquitecturas de las redes troncales [1]. La propia industria ha demostrado que MPLS es una de las tecnologías que están conduciendo la Internet del futuro, aportando un nuevo paradigma de reenvío que afecta, tanto a su capacidad de ingeniería de tráfico, como a la implementación de *Virtual Private Networks* (VPN) [2].

MPLS es una tecnología para el reenvío mejorado de paquetes a través de la red, empleando la información contenida en una pequeña cabecera o etiqueta que se inserta, o bien entre la cabeceras de Capa 3 y de Capa 2 para el caso de tecnologías de Capa de Enlace basadas en paquetes, o bien utilizando los campos

Virtual Path Identifier (VPI) y *Virtual Channel Identifier* (VCI), para el caso de tecnologías basadas en celdas, como ATM. En este sentido, MPLS combina tecnologías de conmutación de Capa 2 con otras de encaminamiento, propias de Capa 3, con el objetivo de buscar un rendimiento y estabilidad mejorados [3]. Para ello MPLS soporta la creación de diferentes rutas entre un nodo fuente y un destino sobre troncales de Internet, basándose estrictamente en el encaminamiento de paquetes, pero aportando técnicas de ingeniería de tráfico y VPN, las cuales ofrecen calidad de servicio (QoS) con múltiples clases de servicio [4]. De esta forma los proveedores de servicio disponen de más técnicas para afrontar el desafío del crecimiento exponencial de uso de la red, mientras tienen la oportunidad de diferenciar servicios sin sacrificar la infraestructura de red existente [5].

Como tecnología de conmutación de etiquetas, MPLS permite a los nodos intermedios tomar decisiones de reenvío basadas en el contenido de una etiqueta simple, en lugar de realizar búsquedas de rutas complejas basadas en el destino de los paquetes, lo que aporta ciertas ventajas:

- Integración IP/ATM: Tradicionalmente, la mayoría de las redes de transporte empleaban un modelo superpuesto, con ATM en la Capa 2 e IP en la Capa 3. Sin embargo, estas implementaciones tienen graves problemas de escalabilidad [6]. Con MPLS los proveedores han migrado muchas de las funciones del plano de control de ATM a la Capa 3, simplificando por tanto el aprovisionamiento de red y su gestión y complejidad. Así se aporta mayor escalabilidad y se elimina al mismo tiempo el inherente *overhead* propio de las celdas ATM al transportar tráfico IP [7].
- En redes privadas virtuales (VPNs): Las VPN clásicas están implementadas como túneles *IP Security* (IPSec) sobre la red pública de Internet pero, aunque cumplen con su cometido, presentan gran *overhead*, son lentas y escalan difícilmente. MPLS implementa VPNs de Capa 3 para múltiples clientes con menor complejidad y coste de aprovisionamiento, operación y mantenimiento [8]. Además, no se requiere cifrado ni software de usuario, ofreciendo por ello un ancho de banda y niveles de servicio comparables a los servicios clásicos de ATM y *Frame Relay*.
- *Traffic Engineering* (TE): Las tecnologías convencionales con ingeniería de tráfico fuerzan a los paquetes a utilizar las rutas que están siendo infrautilizadas para repartir la carga de la red. Estos caminos se seleccionan explícitamente cuando los paquetes entran en la red y su identidad debe

transportarse en ellos, con el consiguiente *overhead* que esto supone [9]. MPLS, en su lugar, proporciona la capacidad de configurar rutas explícitas simples o múltiples, pero empleando la etiqueta para representar la ruta, por lo que la identidad de la ruta explícita no necesita transportarse con el paquete. Esto da lugar a una ingeniería de tráfico más eficiente, permitiendo, como en las técnicas convencionales de balanceo de carga, descargar enlaces congestionados y repartir ese exceso de tráfico por otras rutas que están siendo infrautilizadas [10]. Esta eficiencia se traduce, indirectamente, en una mayor capacidad de aceptación de nuevas conexiones, o en un mayor grado de uso de recursos.

- *Quality of Service* (QoS): Con MPLS los proveedores de servicio pueden proporcionar múltiples *Class of Service* (CoS), incluso con estrictas garantías de QoS, a sus clientes. Para ello asignan reglas diferentes a cada CoS, con el objetivo de priorizar unos flujos con respecto a otros o mejorar métricas básicas, como el *delay* de los flujos de datos [11]. Al mismo tiempo, éstos se pueden tarificar en función del ancho de banda contratado u otros parámetros [12].

2.2 Clases de Equivalencia de Reenvío

Forwarding Equivalence Class (FEC) es la técnica empleada para clasificar el tráfico y da lugar a la capacidad de provisión de QoS ofrecida por MPLS. Así, todos los paquetes que pertenecen a un mismo FEC se reenvían siguiendo criterios similares o a través de la misma ruta. Esta agrupación de paquetes en clases se puede utilizar para establecer prioridades y dar soporte a operaciones eficientes de QoS [13]. Por ejemplo, se pueden asociar FEC de mayor prioridad con servicios en tiempo real, de baja prioridad con acceso web, etc.

La FEC es una entidad lógica creada por el router a través del cual un nuevo flujo de datos va a entrar al dominio y se emplea para representar o categorizar paquetes. Cuando un nuevo paquete llega a este router se analiza su cabecera, para comprobar a qué FEC de las ya predefinidas por el router se ajusta mejor. La FEC elegida dictaminará la etiqueta de salida que se debe añadir al paquete, así como la ruta a seguir.

Básicamente, el router crea una nueva FEC en función de los destinos que conoce, según su tabla de encaminamiento. Por tanto, a priori el router podría crear una FEC diferente para cada destino. En este caso se hablaría de *granulado fino*, en el que una FEC sólo incluiría, por ejemplo, los paquetes con el mismo

destino, los que posean el mismo valor en el campo *Type of Service* (ToS) de su cabecera o sólo los generados a partir de la misma aplicación entre equipos finales. Así, esta gran diferenciación de flujos permite un control muy eficiente de las operaciones con requisitos de QoS, pero como contrapartida se genera un mayor coste de procesamiento debido al elevado número de FEC definidas, necesitándose una mayor y más compleja tabla de encaminamiento en los routers [14]. En el otro extremo, si por ejemplo, el router considera FEC en las que se incluyen todos los paquetes cuya dirección destino tiene un determinado prefijo, se hablaría de *granulado grueso*. Habría un mayor ahorro de etiquetas asignadas, mayor velocidad de procesamiento y el sistema escalaría mejor. El inconveniente es que el grado de diferenciación de tipos de tráfico sería menor y las operaciones de QoS podrían estar más limitadas. La estrategia a seguir dependerá, por tanto, del tipo de tráfico que va a entrar al dominio MPLS, buscando siempre el compromiso entre la escalabilidad y los requisitos de QoS demandados por los flujos de datos.

Es importante destacar que para llevar a cabo la asignación de FEC, un nodo MPLS no sólo utiliza la dirección destino del paquete, sino que puede emplear otros campos de la cabecera o incluso información que no obtiene de la cabecera de la Capa de Red como, por ejemplo, el puerto de entrada o la dirección física de la interfaz por la que llega el paquete. Además, un paquete que entra al dominio MPLS a través de un determinado router se puede etiquetar de forma diferente que si entrara a través de otro router. Esto también facilita la asignación de FEC cuando el reenvío depende del router de entrada a la red [15]. Esta estrategia no es viable mediante el reenvío convencional de Capa 3, ya que la identidad del router a través del cual entra un paquete a la red no se transporta con el paquete. Toda esta funcionalidad ofrecida por el concepto de FEC permite una administración más sencilla de la QoS ofrecida por MPLS, que puede ser total o parcialmente inferida a partir de la FEC y por ende de la etiqueta asociada a los paquetes.

2.3 Conmutación de etiquetas

La etiqueta es una pequeña cabecera de tamaño fijo, que los dispositivos del borde de un dominio de conmutación de etiquetas añaden a los paquetes cuando éstos entran en el mismo, eliminándola cuando salen. El resto de conmutadores (intermedios) utilizarán la etiqueta de los paquetes, para determinar cómo y por dónde se deben reenviar. Es un valor numérico que resume la información esencial acerca de la transmisión de un paquete. Codifica la información relacionada con el destino de los datos, su prioridad, identificación VPN, información de QoS o la ruta basada en ingeniería de tráfico para el paquete. Tiene una longitud fija de 32 bits y actúa como identificador de FEC, es decir, la etiqueta adjunta a un paquete particular representa al FEC al cual se ha asignado ese paquete.

Para el caso de ATM, la etiqueta puede utilizar los campos *Virtual Path Identifier* (VPI) o *Virtual Circuit Identifier* (VCI) de la cabecera ATM. Si, en cambio, el paquete fue generado por *Frame Relay*, la etiqueta puede ocupar el campo *Data Link Connection Identifier* (DLCI). Otras tecnologías de Capa 2, como *Ethernet*, *Token Ring*, FDDI o enlaces punto a punto no pueden utilizar sus cabeceras de nivel 2 para transportar etiquetas [16]. En su lugar transportan cabeceras intermedias insertadas entre la Capa de Enlace y la de Red. El empleo de esta cabecera intermedia para transportar la etiqueta extiende el soporte de MPLS a la gran mayoría de tecnologías de Capa 2, aunque también puede transportar otros protocolos de Nivel 3, como IPv6, IPX o AppleTalk. Esta propiedad ha hecho que MPLS esté ayudando a la migración IPv4 a IPv6.

La Figura 2-1 muestra el formato de cabecera MPLS, la cual contiene los siguientes campos:

- *Etiqueta* (20 bits). Transporta el valor actual de etiqueta MPLS. Es el campo de mayor longitud y ha provocado que a la cabecera MPLS también se la conozca comúnmente como *Etiqueta MPLS*.
- *Clase de servicio* (3 bits). Afecta a los algoritmos de encolado y descarte aplicados al paquete durante su reenvío a través de la red.
- *Pila de etiquetas* (1 bit). El bit *Pila* da soporte al apilado jerárquico de etiquetas MPLS, para permitir la adición de varias cabeceras MPLS al paquete IP [17]. En particular, el bit pila de la etiqueta del fondo de la pila se establece a 1 para indicar que se trata de la última etiqueta y el del resto se establece a 0. Se utiliza, por ejemplo, cuando un paquete entra en un nuevo dominio MPLS sin haber salido antes del anterior, es decir, que el

nuevo es un subdominio integrado en el anterior. A la entrada del subdominio se añadiría una nueva etiqueta a la cima (con el bit pila establecido a 0) y a la salida se extraería. El reenvío de paquetes siempre se lleva a cabo empleando los valores de etiqueta de la cima de la pila.

– *Time To Live* (TTL) (8 bits). El campo TTL es similar al transportado en la cabecera IP, teniendo funcionalidad similar [18].

Los métodos de reenvío de paquetes convencionales pueden llegar a utilizar múltiples algoritmos para el reenvío *unicast*, *multicast* o con requisitos de QoS [19]. Sin embargo, el mecanismo de reenvío de MPLS sólo está basado en la conmutación de etiquetas. Para ello, cada nodo MPLS mantiene dos tablas importantes relacionadas con el encaminamiento de paquetes: la *Label Information Base* (LIB) y la *Label Forwarding Information Base* (LFIB). La primera es una relación de las etiquetas asignadas localmente a paquetes por parte del nodo MPLS, así como la asociación de estas etiquetas a las solicitadas por sus nodos vecinos. La LFIB utiliza un subconjunto de estas etiquetas de la LIB para el encaminamiento de paquetes y consiste en una secuencia de filas o entradas, como se muestra en la Tabla 2-1. Cada fila incluye un valor numérico de etiqueta entrante que la identifica o indexa y una o varias subentradas en función de si se trata de tráfico *unicast* o *multicast*, respectivamente.

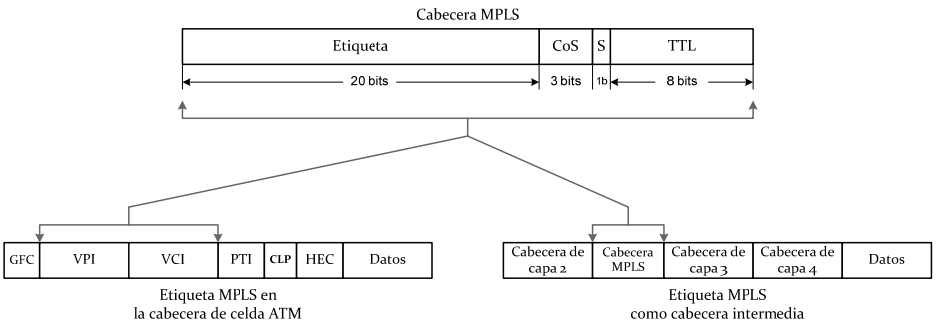

Figura 2-1. Formato de etiquetas MPLS

Tabla 2-1. Formato de LFIB

	Subentrada 1	**Subentrada 2**	**Subentrada 3**	**...**
Etiqueta entrante	Etiqueta de salida Interfaz de salida Dir. del sig. salto	Etiqueta de salida Interfaz de salida Dir. del sig. salto	Etiqueta de salida Interfaz de salida Dir. del sig. salto	...
Etiqueta entrante	Etiqueta de salida Interfaz de salida Dir. del sig. salto	Etiqueta de salida Interfaz de salida Dir. del sig. salto	Etiqueta de salida Interfaz de salida Dir. del sig. salto	...
Etiqueta entrante	Etiqueta de salida Interfaz de salida Dir. del sig. salto	Etiqueta de salida Interfaz de salida Dir. del sig. salto	Etiqueta de salida Interfaz de salida Dir. del sig. salto	...
...	...	...	...	

Cada subentrada incluye etiqueta saliente, interfaz de salida y dirección del siguiente salto. Las subentradas de una misma fila pueden tener las mismas o diferentes etiquetas de salida. El reenvío multicast puede requerir subentradas con múltiples etiquetas de salida. Así, un paquete entrante que llega a una interfaz podría enviarse hacía múltiples interfaces de salida. Además, cada entrada puede incluir información relacionada con los recursos que el paquete puede usar, como la cola de salida en la que el paquete se debería colocar.

MPLS, por tanto, basa el reenvío de paquetes en el intercambio de etiquetas. Primero los nodos leen la etiqueta de los paquetes entrantes y utilizan este valor para buscar alguna coincidencia en la LFIB. Cuando se encuentra una entrada cuyo valor *etiqueta entrante* coincide con el de la etiqueta del paquete, el nodo MPLS sustituye la del paquete por el valor *etiqueta saliente* de la subentrada correspondiente de la LFIB y envía el paquete a través de la interfaz de salida especificada hacia el siguiente salto indicado en la subentrada y, si además se especifica una cola de salida, coloca el paquete en la cola indicada. Igualmente, si el nodo MPLS mantiene múltiples LFIBs, utilizará la interfaz física por la que llegó el paquete para seleccionar una LFIB particular, la cual será la que se consulte para reenviar el paquete [20].

El *Label Switching Router* (LSR) es el dispositivo encargado de reenviar el tráfico basándose en el valor de la etiqueta insertada en los paquetes. Un paso fundamental en la conmutación de etiquetas es que los LSRs previamente

acuerden las etiquetas que deberían utilizar para reenviar el tráfico. Para esta tarea emplean protocolos de distribución de etiquetas. Los nodos del borde del dominio MPLS se denominan *Label Edge Router* (LER) y son los que aplican la etiqueta a los paquetes. Realizan la asignación (*push*) de etiquetas si se encuentran a la entrada del dominio MPLS, o la extracción (*pop*) si se encuentran a la salida.

En la Figura 2-2 se ilustra un ejemplo básico de encaminamiento *unicast* de un paquete sin utilizar pila de etiquetas. En esta topología LSR1 realiza la función de LER, aplicando la etiqueta inicial al paquete después de asignarle un FEC en función de su cabecera IP. Esta asignación se realiza sólo una vez, al entrar en el dominio MPLS.

En este ejemplo, LSR1 recibe un paquete sin etiquetar, al cual añade una etiqueta con valor 7, reenviándolo luego por la interfaz 1 hacia LSR2. Al recibir el paquete éste lee la etiqueta entrante, que tiene valor 7, consulta la LFIB y descubre que debe conmutar la etiqueta 7 por el valor 8, reenviándolo luego por la interfaz 2 hacia LSR4. LSR4 es el LER de salida, consulta su LFIB y descubre que debe extraer la etiqueta. Una vez que el paquete no esté etiquetado, el nodo debe actuar como un router de Nivel 3, consultando su tabla de encaminamiento para decidir cuál es el siguiente salto hacia el destino del paquete. Por tanto LSR4 debe hacer una doble búsqueda.

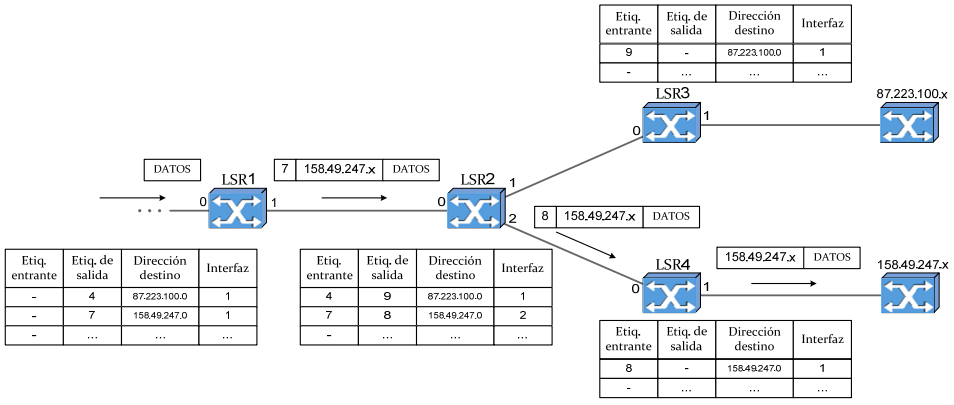

Figura 2-2. Encaminamiento MPLS basado en conmutación de etiquetas

Primero necesita examinar la etiqueta del paquete para luego buscar una coincidencia en su LFIB y sólo averiguar que debe extraer la etiqueta. Después debe hacer una segunda búsqueda, esta vez de Nivel 3, para poder reenviar el paquete hacia el router externo al dominio MPLS. Esta doble búsqueda da lugar, a una degradación del rendimiento, así como a una mayor complejidad de implementación. Para evitar estos inconvenientes, MPLS implementa una técnica de extracción de etiquetas en el penúltimo salto, en la que el nodo encargado de extraer la etiqueta es el vecino previo a LSR4. Así, el último nodo recibirá el paquete ya sin etiqueta, por lo que sólo tendrá que hacer una búsqueda, la de Capa 3 basada en la dirección destino del paquete, para luego reenviarlo hacia el siguiente salto externo.

El conjunto de nodos MPLS por los que pasan los paquetes etiquetados se denomina *Label Switched Path* (LSP) y, como ya se ha comentado, es una conexión configurada entre 2 LER del dominio. Cada LSP se elige en función de los requisitos del flujo de datos que vaya a utilizarlo. De hecho, el LSP se suele definir como la ruta o conjunto de LSRs que atraviesan los paquetes pertenecientes a un cierto FEC cuando viajan hacia su destino.

El establecimiento o señalización del LSP es un paso previo al reenvío de paquetes. Su correcta configuración es imprescindible, ya que, durante este proceso, se lleva a cabo el reparto de etiquetas, es decir, una recomendación que cada nodo hace a su vecino previo acerca de la etiqueta que debe colocar en los paquetes que le envíe para cada FEC particular. Esta asignación de etiquetas está controlada de forma ordenada desde el nodo de salida hacia el de entrada del LSP, que es donde se ha hecho la solicitud de creación del LSP. Este es un proceso de *control ordenado*, que requiere que la elección de etiquetas se propague por todos los LSRs y en el que cada nodo dicta al anterior cuál debe ser la etiqueta saliente de los paquetes que le envíe. Esto resulta en unos tiempos de convergencia mayores a los que se conseguirían empleando un *control independiente*, en el que las elecciones de etiquetas las toma cada nodo independientemente, sin atender las indicaciones de otros nodos. MPLS soporta ambos mecanismos de señalización de rutas, aunque el control ordenado ha demostrado mejores capacidades de prevención de bucles en el LSP que el control independiente, el cual podría no ser adecuado para redes de topologías complejas o extensas por este motivo.

2.4 MPLS-Traffic Engineering

Traffic Engineering (TE) es un nombre genérico que corresponde al uso de diferentes técnicas para optimizar el uso de la capacidad y topología de una red troncal. MPLS-TE ha proporcionado una forma de integrar capacidades de TE típicas de protocolos de Capa 2 como ATM en protocolos de Capa 3 como IP. Sin embargo, el conjunto de nodos por los que se configura un LSP o *LSP tunel*, como se suele denominar en el ámbito de MPLS-TE, viene determinado por el compromiso entre los recursos que demandan los flujos de datos y la capacidad real de la red [21]. Esta información acerca de los recursos disponibles puede distribuirse mediante extensiones de protocolos de encaminamiento, como *Open Shortest Path First* (OSPF) o *Intermediate System to Intermediate System* (IS-IS), a los que se les da la capacidad de interpretar ciertas condiciones o restricciones a la hora de encontrar posibles caminos hacia el destino de los paquetes [22]. Así, el algoritmo empleado por estos protocolos para generar una ruta con ingeniería de tráfico es diferente a los algoritmos tradicionales de cálculo de rutas basados en *Shortest Path First* (SPF), ya que se requieren algoritmos capaces de interpretar restricciones, como *Constrained Shortest Path First* (CSPF).

Por un lado, un algoritmo como CSPF no está diseñado para encontrar la mejor ruta hacia el resto de routers, sino sólo hacia el nodo final del túnel LSP o LER de salida del dominio MPLS. Esto diferencia a CSPF, ya que el proceso se detiene tan pronto como el nodo que se está intentando alcanzar se añade al conjunto de nodos del LSP, en lugar de intentar obtener las mejores rutas que lleguen al resto de nodos del dominio, como haría un algoritmo tradicional.

Por otro lado, CSPF también se diferencia en que, además del coste simple de enlace entre cada dos vecinos, también tiene en cuenta otras métricas, como el ancho de banda disponible, algunos atributos de enlace o el peso administrativo de los nodos. Por ejemplo, en la Figura 2-3 se quiere señalizar un túnel con ingeniería de tráfico desde el router A hasta el D, con un requisito de ancho de banda de 60 Mbps. Cada enlace de la figura muestra su coste y ancho de banda disponible. Sin tener en cuenta este último parámetro, la mejor ruta desde el router A hasta el router D es A-B-C-D, con un coste total de 12. Sin embargo, esta posible ruta no dispone de 60 Mbps en todos sus nodos, por lo que se debe desechar. Por este motivo CSPF debe conocer los parámetros de todos los nodos de la ruta, no sólo los del siguiente salto.

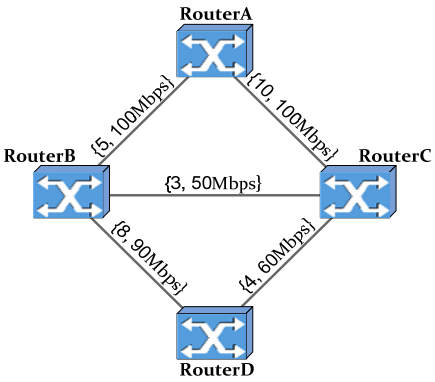

Figura 2-3. Funcionamiento básico del algoritmo CSPF

El algoritmo SPF común (tal y como se emplea en OSPF o IS-IS) es válido para elegir múltiples rutas con el mismo coste hacia el destino, lo que se conoce como *Equal-Cost MultiPath* (ECMP). No obstante, si para CSPF se encuentran varias rutas que cumplan todas las restricciones demandadas por el flujo, se deben emplear parámetros adicionales para desempatar, es decir, para encontrar la que mejor se ajuste a los requisitos demandados. Los pasos que sigue CSPF en caso de empate entre dos o más rutas son los siguientes:

1º Elegir la ruta con el mayor ancho de banda mínimo disponible.

2º Si todavía existe empate, tomar la ruta con menor número de saltos.

3º Si persiste el empate, elegir una de ellas al azar.

En el dominio de la Figura 2-4 se pretende crear un túnel desde el router A hasta el Z, con una demanda de ancho de banda de 10 Mbps. Existen cinco posibles rutas desde A a Z (identificadas de arriba a abajo como RUTA1 a RUTA5). Todas disponen del suficiente ancho de banda para cumplir con la demanda. El proceso de decisión que sigue CSPF para elegir la mejor ruta determina que RUTA1 no es elegible porque tiene un coste más elevado que las otras. RUTA2 tampoco es la óptima porque su ancho de banda mínimo es 80 Mbps, que es menor que el de las otras. RUTA3 presenta un mayor número de saltos y RUTA4 o RUTA5 se pueden elegir por igual.

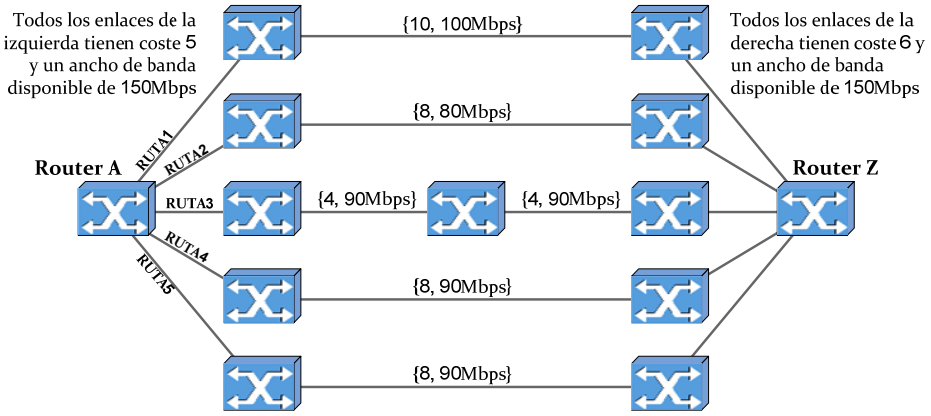

Figura 2-4. Proceso de desempate del algoritmo CSPF

Después de que el algoritmo CSPF haya obtenido la mejor ruta en función de los requisitos del flujo de datos, el camino debe señalizarse a través de la red, para así configurar una relación nodo a nodo que represente la ruta. En este sentido, RSVP-TE o CR-LDP han sido protocolos extensamente utilizados para señalizar rutas en dominios MPLS. Ambos protocolos presentan mecanismos de señalización con soporte de ingeniería de tráfico a través de troncales MPLS, con el objetivo de que la creación de los túneles LSP no quede limitada a los mecanismos clásicos de selección de rutas IP. Así, ambos ofrecen la capacidad de extender o repartir el tráfico de datos más uniformemente a través de la red, teniendo en cuenta todos los enlaces disponibles. En cierto modo, RSVP-TE es un protocolo de señalización con QoS estandarizado por el IETF, ofreciéndose como una extensión de RSVP para el soporte de distribución de etiquetas y encaminamiento explícito [23], mientras que CR-LDP se propuso como extensión de LDP, también orientado a la distribución de etiquetas, con soporte de señalización de QoS y encaminamiento explícito [24].

Existen otras muchas similitudes entre ambas propuestas, como que los objetos utilizados para señalizar rutas explícitas son muy similares, ambas utilizan procedimientos de control ordenado para la configuración de LSP, o que ambas incluyen información sobre QoS en los mensajes de señalización, para la localización de recursos y para el establecimiento automatizado de LSP.

Como diferencias entre ambos protocolos encontramos que RSVP-TE es un protocolo que se denomina de *estado blando*, pues necesita refrescar de forma

periódica las rutas señalizadas, ya que los mensajes RSVP-TE se encapsulan en datagramas IP ó UDP, que son protocolos que no aportan fiabilidad a la comunicación. En su lugar, CR-LDP utiliza TCP para la distribución de etiquetas y no requiere refrescar periódicamente la información, porque TCP ya aporta fiabilidad a la comunicación. En cierta medida, esto hace que CR-LDP sea un poco más lento que RSVP-TE. Además, sólo RSVP-TE proporciona notificaciones nativas ante la ocurrencia de fallos en el dominio.

Por consiguiente, tanto CR-LDP como RSVP-TE se han utilizado como protocolos de señalización en redes MPLS, realizando funciones muy similares. De hecho, tradicionalmente nunca ha existido consenso sobre qué protocolo era técnicamente superior. Hasta cierto punto, RSVP-TE fue adoptado por los principales fabricantes de dispositivos MPLS por su sencilla integración en las redes IP que ya tenían desplegadas. Además, el IETF decidió centrarse exclusivamente en RSVP-TE [25], dejando obsoleto CR-LDP.

También es importante destacar que RSVP-TE se basa en RSVP, que es un protocolo tradicionalmente utilizado para la reserva de recursos en redes clásicas [26]. Si bien, se debe tener en cuenta que RSVP-TE no es un protocolo de encaminamiento, por lo que cualquier decisión de encaminamiento la deben tomar previamente el *Interior Gateway Protocol* (IGP) y CSPF. La tarea de RSVP-TE se centra en la señalización y mantenimiento de las rutas y de las reservas de recursos que requiera cada ruta. En particular, en MPLS-TE se habla de una arquitectura de dos planos, el Plano de Control y el Plano de Reenvío. El primero es el ámbito en el que RSVP-TE crea túneles LSP, para los que además intenta reservar un ancho de banda suficiente como para cubrir a priori las necesidades del flujo de datos que utilizará esa ruta. El Plano de Reenvío es el ámbito en el que el flujo de datos utiliza la ruta señalizada por RSVP-TE, el cual se debe encargar de mantener. En el Plano de Control se rellena la LFIB, mientras que en el Plano de Reenvío el nodo MPLS lleva a cabo el encaminamiento del tráfico, basándose en el valor de la etiqueta adjunta a cada paquete, como se ha detallado anteriormente. Para ello utiliza la tabla LFIB, ya rellena, así como la información inferida a partir del valor de la etiqueta, para conmutar etiquetas y reenviar paquetes hacia el siguiente salto del túnel LSP. Por otra parte, aunque existen propuestas de re-optimización de rutas para balancear el tráfico [27], RSVP-TE no aporta un mecanismo de vigilancia que impida que un flujo haga un uso mayor de recursos de los que tiene reservados [28].

RSVP-TE es uno de los principales protagonistas del Plano de Control MPLS. En cambio, los nodos también deben ejecutar un protocolo de encaminamiento IP para intercambiar información de encaminamiento con el resto de nodos MPLS de la red. En este sentido, los protocolos de estado de enlace, como OSPF o IS-IS, son los más adecuados, ya que pueden proporcionar a cada nodo MPLS una visión del estado del resto de la red [29], [30]. Así, OSPF distribuye la información de encaminamiento entre nodos no necesariamente adyacentes, mientras que la información de vinculación de etiquetas se distribuye sólo entre nodos adyacentes. Por este motivo los protocolos de encantamiento basados en estado de enlace no son adecuados para distribuir información sobre asignación de etiquetas, por lo que para ello se emplea un protocolo como RSVP-TE.

Los túneles LSP comparten muchas de las características de los circuitos virtuales de ATM. Se configuran y encaminan explícitamente y tienen un rico conjunto de mecanismos de QoS. En particular, el mensaje *Path* es un mensaje RSVP-TE que se envía desde el nodo de entrada al dominio para el flujo de datos en cuestión en dirección al nodo de salida. Permite indicar el camino o conjunto de nodos que darán lugar al túnel LSP, asignando también recursos provisionalmente en los nodos de la ruta. La respuesta es un mensaje *Resv*, que se envía en dirección contraria. Éste hace el reparto final de etiquetas y convierte la asignación provisional en una reserva de recursos permanente. Existen otros tipos de mensajes RSVP-TE, como se muestra en la Tabla 2-2, ya que además de la señalización y mantenimiento de LSP, RSVP-TE también se encarga del cierre de los LSP o de la notificación de errores.

Tabla 2-2. Tipos de mensajes RSVP-TE

Mensaje	**Descripción**
Path	Usado para configurar o mantener reservas.
Resv	Respuesta a un mensaje *Path*.
PathErr	Enviado por el receptor de un mensaje *Path*, si detecta algún error.
ResvErr	Enviado por el receptor de un mensaje *Resv*, si detecta algún error.
Hello	Usado para comprobar la comunicación con nodos vecinos.
PathTear	Análogo al mensaje *Path*, pero usado para el cierre de un LSP.
ResvTear	Análogo al mensaje *Resv*, pero usado para el cierre de un LSP.

El router que envía un mensaje *Path* se suele denominar PHOP (*Previous Hop*) y el que lo recibe NHOP (*Next Hop*). Después de que éste lo reciba primero debe comprobar si el formato del mensaje es correcto. Luego comprueba la reserva de recursos que se solicita. Esto es lo que se denomina como Control de Admisión de MPLS. Si tiene éxito y se permite al mensaje *Path* reservar el ancho de banda que demanda, el NHOP crea un nuevo mensaje *Path* y lo envía a su NHOP siguiendo, como siempre, la ruta calculada por CSPF hacia el destino. Esta cadena de mensajes *Path* continúa hasta que se alcanza el último nodo del túnel, que es el LER de salida del dominio MPLS para el flujo de datos en cuestión. Este router lleva a cabo el control de admisión sobre el mensaje *Path* al igual que el resto de nodos anteriores. Si también tiene éxito, finalmente este nodo contesta a su PHOP con un mensaje *Resv*, que se puede considerar como un mensaje de confirmación para la solicitud de creación del LSP implícita en el mensaje *Path* replicado desde el LER de entrada. El mensaje *Resv* no sólo contiene la confirmación de que se ha hecho una reserva de recursos satisfactoria en todos los nodos del camino, sino que incluye también la etiqueta saliente que su PHOP debe utilizar cuando le envíe paquetes de datos. Al igual que con los mensajes *Path*, la cadena de mensajes *Resv* continúa, pero en este caso en sentido contrario, es decir, cada NHOP envía un mensaje *Resv* a su PHOP confirmando la reserva y solicitando etiqueta, hasta que se alcance el LER de entrada. En el caso de que se produzca algún problema durante la señalización, RSVP-TE también podría notificarlo. Para ello dispone de los mensajes *PathErr* o *ResvErr*. Si un nodo detecta un error en un mensaje *Path* o si comprueba que la demanda de recursos no es factible, responderá con un *PathErr*. Si en cambio detecta un error en un mensaje *Resv*, contestará con un *ResvErr*. Los mensaje *PathErr* se envían de un NHOP a su PHOP y los *ResvErr* en sentido contrario, teniendo como consecuencia en ambos casos el rechazo de la nueva conexión durante el control de admisión.

Una vez que el LSP se ha señalizado con éxito, comienza la transmisión de paquetes etiquetados desde el nodo de entrada al dominio. Aquí puede aparecer el mensaje *Hello*, que es un tipo especial de mensaje RSVP-TE. Se envía entre nodos vecinos durante la transmisión de datos para comprobar la integridad de la comunicación entre ambos. Su uso es opcional y los nodos que no tengan dicha capacidad o que no estén configurados para ello pueden ignorarlos. De hecho la recepción de un mensaje *Hello* no afecta al funcionamiento habitual del nodo. Su funcionamiento es sencillo: cuando un nodo necesite comprobar la conectividad con uno de sus vecinos, le envía un mensaje de solicitud *Hello*. Si durante un intervalo de tiempo predefinido no recibe un mensaje *Hello* de

confirmación desde su vecino, puede determinar que se ha perdido la comunicación con el nodo par. En este caso podría iniciar el proceso de conmutación a un LSP de respaldo, si existe, o avisar del error para proceder al cierre del LSP.

Finalmente, el cierre de un LSP es el proceso que tiene lugar cuando un nodo (normalmente el LER de entrada del túnel) decide que una reserva ya no es necesaria en la red. En ese momento envía un mensaje *PathTear* por la misma ruta que siguió el mensaje *Path*. También recibirá un mensaje *ResvTear* de confirmación para ese mensaje *PathTear* (por la misma ruta que se enviaron los mensajes *Resv*), para indicar que el NHOP ha liberado la reserva de recursos que tenía concertada para ese LSP. Este intercambio de mensajes *Pathtear*/*ResvTear* continúa hasta alcanzar el nodo final del túnel, quedando en ese momento liberados todos los recursos del túnel.

2.5 MPLS Transport Profile

MPLS está considerada como una de las tecnologías orientada a conexión y con soporte de QoS más eficientes para el transporte de paquetes. Este hecho, unido al amplio despliegue existente de MPLS en todo el mundo, está provocando un gran esfuerzo de estandarización de una versión simplificada y especializada de MPLS, pero totalmente compatible con este protocolo [31]. Este estándar es lo que ha pasado a conocerse como *MPLS Transport Profile* o MPLS-TP, propuesto por IETF, en los grupos MPLS, PWE3 y CCAMP y el ITU-T SG15. Es un interés que se debe, principalmente, a que la gran mayoría de redes en explotación están orientadas al transporte de estructuras de tipo *frame*, celda, datagrama, etc. Esto ha dado lugar a la evolución de las clásicas redes de transporte basadas en multiplexación por división en el tiempo hacia nuevas arquitecturas optimizadas para el transporte de paquetes [32]. Básicamente, la función de una red de transporte es la de transmitir la información contenida en estas estructuras de formato definido entre los servicios de los equipos finales. Estos dispositivos pueden ser multiplexadores de acceso a servicios DSL, *gateways*, multiplexadores T1/E1, servidores de acceso remoto a banda ancha, etc. No obstante, en el punto de acceso, en el nodo de agregación de flujos o en dominios metropolitanos, los proveedores de servicio han visto necesaria la simplificación de las comunicaciones basadas en el transporte de paquetes, con el objetivo de reducir, tanto la inversión inicial, como los gastos de gestión de red en sistemas de nueva generación.

MPLS-TP consta de un conjunto de mejoras con respecto al protocolo MPLS original, extendiendo la definición de éste para aportarle una mayor compatibilidad con los mecanismos tradicionales de transporte de paquetes [33]. MPLS-TP hereda todas las técnicas de QoS soportadas por MPLS, así como los mecanismos definidos en diferentes estándares que mejoran la funcionalidad de MPLS. Sin embargo, una de las grandes ventajas que aporta MPLS-TP son los beneficios que proporcionan los mecanismos de protección OAM (*Operations, Administration and Maintenance*), con señalización en banda y que ya se podían encontrar en tecnologías de transporte clásicas [34]. De esta forma, si se produce algún problema en el LSP, los mecanismos OAM deben ser capaces de detectarlo y diagnosticarlo [35]. Deben existir, además, los mecanismos adecuados para que se puedan tomar las acciones correctivas adecuadas. Estos errores pueden deberse a fenómenos tales como la corrupción, duplicidad o desorden de paquetes o las pérdidas en condiciones de trabajo normales de la red (fallos de enlaces o de nodos, descartes por congestión, etc.) [36]. Al mismo tiempo, los mecanismos correctivos deben estar disponibles para cada proveedor de servicios particular, pero no deben verse afectados por los problemas de su servicio. Para ello se propone que las soluciones que se adopten para conseguir estos objetivos deben basarse en el intercambio fiable de información entre nodos, de forma que, por ejemplo, la disminución del rendimiento o los fallos de nodo/enlace se puedan detectar eficientemente, poniendo en conocimiento de los protocolos afectados los necesarios cambios de estado de una manera efectiva [37].

OAM se presenta entonces como una funcionalidad imprescindible en MPLS-TP, ya que contribuye a la mejora de aspectos clave del funcionamiento de MPLS [38], como son:

- La reducción de costes o de la complejidad operacional de la tecnología, permitiendo la detección, diagnóstico, localización y manejo automático y eficiente de las interrupciones del servicio, con el objetivo de minimizar la duración del *downtime*.
- La mejora de la disponibilidad de red, asegurando que los defectos (aquellos sucesos ajenos a averías pero que provocan la no entrega del tráfico al cliente) y los fallos (desconexiones o averías de enlace o nodo) sean detectados, diagnosticados y tratados de forma transparente al cliente.
- Lograr los objetivos de servicio y de rendimiento, ya que la funcionalidad OAM permite la verificación de cumplimiento del SLA en dominios multi-mantenimiento (el llevado a cabo por parte de múltiples nodos, instancias o técnicas diferentes). Esto permite la determinación o medición de la

degradación del servicio en parámetros como, por ejemplo, el retardo o la pérdida de paquetes [39].

En este sentido, una de las herramientas OAM propuestas por el estándar es la medición o monitorización del número de paquetes perdidos, dada la importancia de este parámetro en redes MPLS, sobre todo en lo referente a sus efectos sobre el tráfico perteneciente a servicios preferentes o prioritarios [40]. Así, existen distintas razones que motivan la adición de mecanismos para la detección automatizada de pérdidas de paquetes:

- Algunos tipos de servicios no funcionan correctamente o simplemente no funcionan si la pérdida de datos extremo a extremo supera un determinado umbral.
- Una excesiva pérdida de paquetes puede dificultar el soporte de ciertas aplicaciones de tiempo real, en las que ese valor umbral de pérdida admisible es variable, porque depende de la aplicación en particular.
- Cuanto mayor es el número de paquetes perdidos, más difícil es para los protocolos de capas superiores mantener un rendimiento aceptable.

La sensibilidad de las aplicaciones de tiempo real y de los protocolos de capas superiores a la pérdida de paquetes se convierte, por tanto, en algo clave cuando coexisten un gran número de servicios diferentes pero todos susceptibles al *delay* o al *throughput*, lo cual es muy común es las redes MPLS en explotación actuales y futuras [41].

2.6 GLRP sobre MPLS-TP

Como ya se ha comentado, MPLS ofrece QoS mediante diferenciación del tráfico, pero llevando también a cabo un control de admisión basado en la reserva de recursos de RSVP-TE. Sin embargo, una vez admitido un nuevo flujo de paquetes, el protocolo no proporciona un mecanismo de vigilancia de los recursos consumidos que actúe eficazmente cuando dicho tráfico sobrepase los recursos que se le asignaron inicialmente. De hecho, deja en manos del protocolo de transporte la recuperación de los posibles paquetes perdidos debido a la congestión. El protocolo empleará mecanismos de control de la congestión basados en el *feedback* que proporciona la red. En cierto modo, se debe tener en cuenta que la duración de la congestión está directamente relacionada con el producto *Ancho de Banda * Latencia*. Es decir, cuanto mayor es el retardo extremo a extremo de la red, mayor será el tiempo que necesite el emisor para

decidir que la red se ha congestionado. Pero, al mismo tiempo, cuanto mayor sea el ancho de banda, mayor será la cantidad de datos que el emisor colocará en la red durante el tiempo que necesite para detectar la congestión [42]. Las troncales MPLS/MPLS-TP suelen ser un claro ejemplo de redes con un elevado producto *Ancho de Banda* * *Latencia*. En este sentido, MPLS-TP está desarrollando extensiones sobre MPLS para alcanzar los requisitos de red clásicos, como escalabilidad, multi-servicio, eficiencia de costes, máximo nivel de disponibilidad o capacidades OAM [43]. La monitorización PLM (*Packet Loss Measurement*) es una de estas funciones OAM que se está convirtiendo en un elemento clave para muchos proveedores de servicios, ya que la garantía del SLA ofrecido va a depender, en gran medida, de la capacidad de la red para monitorizar métricas como la pérdida de paquetes o el retardo de los paquetes [44]. Así, GLRP trabaja en dominios MPLS-TP como una nueva función OAM para la mejora de la pérdida de paquetes de servicios con requisitos de QoS. Para ello emplea el propio protocolo de señalización de MPLS para la señalización del entorno operativo necesario, así como de la información relacionada con las pérdidas producidas. A diferencia de los esquemas centralizados, en los que sólo los nodos de los extremos son los encargados de diseminar la información relacionada con las pérdidas, GLRP define un conjunto de nodos cercanos entre los que se distribuye esta información, con el objetivo de recuperar los datos perdidos más rápidamente. Así, GLRP actúa como un mecanismo de monitorización de pérdidas de rápida actuación, retransmitiendo localmente los datos perdidos de flujos prioritarios con requisitos de QoS. Por tanto, GLRP introduce una técnica de control de pérdidas de datos prioritarios en capas inferiores, en particular en el Plano MPLS, con el objetivo de hacer una detección más eficiente. Además, GLRP dará lugar a retransmisiones locales entre nodos cercanos, con el objetivo de mejorar también la latencia de las retransmisiones de los paquetes perdidos.

2.7 Referencias del capítulo

[1] R. Winter, M. Vigoureux and S. Bryant, “The coming of age of MPLS,” IEEE Communications Magazine, vol 49, nº 4. Pág. 78-81. 2011. [Online]: http://dx.doi.org/10.1109/MCOM.2011.5741150

[2] E. Rosen and Y. Rekhter, “BGP/MPLS IP Virtual Private Networks (VPNs),” IETF Request for Comments 4364, Standards Track. 2006.

[3] E. Rosen, A. Viswanathan and R. Callon, “Multiprotocol label switching architecture,” IETF Request for Comments 3031, Standards Track. 2001.

[4] F. Le Faucheur et al., "Multi-Protocol Label Switching (MPLS) support of Differentiated Services," IETF RFC 3270, Standards Track. 2002.

[5] C. Awad, B. Sanso and A. Girard, "Design of Reliable Communication Networks," 7th International Workshop on the Design of Reliable Communication Networks. Pág. 265-272. 2009. [Online]: http://dx.doi.org/10.1109/DRCN.2009.5339998 (último acceso abril 2014).

[6] A. G. Malis, "Converged services over MPLS," IEEE Communications Magazine, vol 44, nº 9. Pag 150-156. 2006. [Online]: http://dx.doi.org/10.1109/MCOM.2006.1705992 (último acceso abril 2014).

[7] M. Bocci and J. Guillet, "ATM in MPLS-based converged core data networks," IEEE Communications Magazine, vol 41, nº 1. Pag 139-145. 2003. [Online]: http://dx.doi.org/10.1109/MCOM.2003.1166672 (último acceso abril 2014).

[8] I. Minei and P.R. Marques, "Scalability Considerations in BGP/MPLS IP VPNs," IEEE Comm. Magazine, vol 45, nº 4. Pag 26-31. 2006. [Online]: http://dx.doi.org/10.1109/MCOM.2007.343608 (último acceso abril 2014).

[9] S. Ladhe, S. Devane and M. Chatterjee, "Improved agent based explicit *Path* computation algorithm for MPLS-TE," 16th IEEE International Conference on Networks, ICON 2008. Pág. 1-5. 2008. [Online]: http://dx.doi.org/10.1109/ICON.2008.4772636 (último acceso abril 2014).

[10] A. Fumagalli, M. Tacca, Kai Wu and J.P. Vasseur, "Local recovery solutions from multi-link failures in MPLS-TE networks with probable failure patterns," IEEE Global Telecommunications Conference, GLOBECOM '04, vol 3. Pág. 1490-1494. 2004. [Online]: http://dx.doi.org/10.1109/GLOCOM.2004.1378230 (último acceso abril 2014).

[11] J. Carmona-Murillo, J.L. González-Sánchez, D. Cortés-Polo and F.J. Rodríguez-Pérez, "DM3: distributed mobility management in MPLS-based access networks," International Journal of Network Management, vol 24, nº 2. Pág. 85-100. 2014. [Online]: http://dx.doi.org/10.1002/nem.1854 (último acceso abril 2014).

[12] R. Prabagaran and J.B Evans, "Experiences with class of service (CoS) in IP/MPLS networks," 26th Annual IEEE Conference on Local Computer Networks, LCN '01. Pág. 243-249. 2001. [Online]: http://dx.doi.org/10.1109/LCN.2001.990793 (último acceso abril 2014).

[13] A.M. Bongale, N. Nithin and L.S. Jyothi, "Traffic prioritization in MPLS enabled OSPF network," World Congress on Information and Communication Technologies, WICT 2012. Pág. 132-137. 2012. [Online]: http://dx.doi.org/10.1109/WICT.2012.6409063 (último acceso abril 2014).

[14] Yan Jiangzhou and Liu Zengji, "Resource allocation and admission control based on flow congestion probability in MPLS networks," 11th International Conference on Advanced Communication Technology, ICACT '09, vol 1. Pág. 694-697. 2009.

[15] Ying-Xiao Xu and Gen-Du Zhang, "Models and algorithms of QoS-based routing with MPLS traffic engineering," 5th IEEE International Conference on High Speed Networks and Multimedia Comm. Pág. 128-132. 2002. [Online]: http://dx.doi.org/10.1109/HSNMC.2002.1032561 (último acceso abril 2014).

[16] L. Martini, E. Rosen, N. El-Aawar and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks," IETF Request for Comments 4448, Standards Track. 2006.

[17] E. Rosen et al., "MPLS label stack encoding," IETF Request for Comments 3032, Standards Track. 2001.

[18] P. Agarwal and B. Akyol, "Time To Live (TTL) processing in Multi-Protocol Label Switching (MPLS) networks," IETF Request for Comments 3443, Standards Track. 2003.

[19] S. Blake et al., "An Architecture for Differentiated Services," IETF Request for Comments 2475, Standards Track. 1998.

[20] G. Armitage, "MPLS: the magic behind the myths [multiprotocol label switching]," IEEE Communications Magazine, vol 38, nº 1. Pag 124-131. 2000. [Online]: http://dx.doi.org/10.1109/35.815462 (último acceso abril 2014).

[21] A. Autenrieth and A. Kirstadter, "RD-QoS - the integrated provisioning of resilience and QoS in MPLS-based networks," IEEE International Conference on Communications, ICC '02, vol 2. Pág. 1174-1178. 2002. [Online]: http://dx.doi.org/10.1109/ICC.2002.997035 (último acceso abril 2014).

[22] K. Shuaib and F. Sallabi, "Extending OSPF for large scale MPLS networks," IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication. Pág. 13-16. 2005. [Online]: http://dx.doi.org/10.1109/SARNOF.2005.1426500 (último acceso abril 2014).

[23] D. Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF Request for Comments 3209, Standards Track. 2001.

[24] B. Jamoussi et al., "Constraint-Based LSP Setup using LDP," IETF Request for Comments 3212, Standards Track. 2002.

[25] L. Andersson and G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols," IETF Request for Comments 3468, Standards Track. 2003.

[26] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP)Version 1, Functional specification," IETF Request for Comments 2205, Standards Track. 1997.

[27] K. Long, Z. Zhang and S. Cheng, "Load balancing algorithms in MPLS traffic engineering," IEEE Workshop on High Performance Switching and Routing. Pág. 175-179. 2001. [Online]: http://dx.doi.org/10.1109/HPSR.2001.923627 (último acceso abril 2014).

[28] M. Arumaithurai, R. Geib, R. Rex and Xiaoming Fu, "Pre-congestion notification-based flow management in MPLS-based DiffServ networks," 28th IEEE International Performance Computing and Comm. Conference, IPCCC '09. Pág. 57-64. 2009. [Online]: http://dx.doi.org/10.1109/PCCC.2009.5403827 (último acceso abril 2014).

[29] D. Katz, K. Kompella and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2," IETF Request for Comments 3630, Standards Track. 2003.

[30] J. Parker, "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)," IETF Request for Comments 3719, Standards Track. 2004.

[31] M. Bocci, M. Vigoureux and S. Bryant, "MPLS Generic Associated Channel," Request for Comments, Junio 2009.

[32] M. Bocci et al., "A Framework for MPLS in Transport Networks," IETF Request for Comments 5921, Standards Track. 2010.

[33] B. Niven-Jenkins et al., "MPLS-TP Requirements," IETF Request for Comments 5654, Standards Track. 2009.

[34] I. Busi and D. Allan, "Operations, Administration and Maintenance Framework for MPLS-Based Transport Networks," IETF Request for Comments 6371, Standards Track. 2011.

[35] M. Vigoureux, D. Ward and M. Betts, "Requirements for Operations, Administration and Maintenance (OAM) in MPLS Transport Networks," IETF Request for Comments 5860, Standards Track. 2010.

[36] G. Swallow et al., "MPLS Fault Management Operations, Administration and Maintenance (OAM)," IETF RFC 6427, Standards Track. 2011.

[37] C. Cao, Y. Zhang, J. Zhang, X. Cheng and W. Gu, "Packet-Level Optimization for Transmission Performance Improvement of Internet-Bound Traffic in a MPLS-TP Network," IEEE/OSA Journal of Optical Communications and Networking, vol 2, nº 11. Pag 991-999. 2000. [Online]: http://dx.doi.org/10.1364/JOCN.2.000991 (último acceso abril 2014).

[38] T. Oishi, M. Takase, K. Sakamoto and H. Endo, “Implementation of packet transport system using MPLS-TP technologies,” 8th Asia-Pacific Symposium on Information and Telecomm. Technologies, APSITT '10. Pág. 1-6. 2010.

[39] D. Cortés-Polo, J.L. González-Sánchez, F.J. Rodríguez-Pérez and J. Carmona-Murillo, “Mobility management in packet transport networks for network convergence,” Transactions on Emerging Telecommunications Technologies, vol [pendiente], nº [pendiente]. Pag [pendiente]. 2013. [Online]: http://dx.doi.org/10.1002/ett.2705 (último acceso abril 2014).

[40] D. Frost and S. Bryant, “A Packet Loss and Delay Measurement Profile for MPLS-based Transport Networks,” IETF Request for Comments 6375, Standards Track. 2011.

[41] U.M. Mir, A.H. Mir, A. Bashir and M.A. Chishti, “DiffServ-aware Multi Protocol Label Switching based quality of service in Next Generation Networks,” IEEE International Advance Computing Conference, IACC '14. Pág. 233-238. 2014.

[42] Y. Li Y, D. Leith D and R.N. Shorten, "Experimental evaluation of TCP protocols for high-speed networks," IEEE/ACM Transactions on Networking, vol. 15, nº 5. Pág. 1109-1122. 2007. [Online]: http://dx.doi.org/10.1109/TNET.2007.896240 (último acceso junio 2014).

[43] T. Oishi, M. Takase, K. Sakamoto and H. Endo."Implementation of packet transport system using MPLS-TP technologies," 8th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT). Pág. 1-6. 2010.

[44] M. Kim, M.W. Mutka and H.Y. Kim."ESC: estimation of selecting core for reducing multicast delay variation under delay constraints," International Journal of Communication Systems, Vol. 24, nº 1. Pág. 40-52. 2011. [Online]: http://dx.doi.org/10.1002/dac.1137 (último acceso julio 2014).

Capítulo 3. GLRP: Gossip-based Local Recovery Policy

La originalidad consiste en el retorno al origen; así pues, original es aquello que vuelve a la simplicidad de las primeras soluciones.
Antoni Gaudí

En el anterior capítulo se abordaron los aspectos más relevantes de MPLS, MPLS-TE y MPLS-TP, incidiendo sobre sus capacidades de ingeniería de tráfico, control de la congestión y QoS. También se elaboró una breve discusión sobre cómo el algoritmo CSPF puede colaborar a la hora de encontrar la mejor ruta entre dos nodos, en función de los requisitos de QoS del nuevo flujo de datos. En este capítulo se definirá nuestra propuesta GLRP, explicando con detalle su misión, así como la integración con el protocolo de señalización RSVP-TE. También se analizará la arquitectura del nodo con capacidad GLRP.

3.1 Definición de GLRP

Es habitual que los routers de un dominio de red MPLS se vean sometidos a congestiones temporales debidas a la propia impredecibilidad del tráfico, lo cual puede provocar pérdidas de datos [1]. Supongamos, por ejemplo, un nodo perteneciente a una ruta entre una fuente y un destino. Si en un instante dado este nodo es incapaz de reenviar un paquete hacia el siguiente salto, esto provocará la pérdida de dicho paquete. El protocolo de transporte, el cual aporta fiabilidad a la comunicación, iniciaría de nuevo el envío de los datos perdidos desde el origen, pero sólo a partir del momento en que detecte la pérdida de tráfico. Si se trata de un servicio con elevados requisitos de retardo y fiabilidad,

los procesos de detección y de retransmisión extremo a extremo de los datos perdidos afectarían negativamente a las métricas relacionadas con el retardo de los paquetes [2].

Como se ha comentado en secciones anteriores, el protocolo de transporte deja la detección de las pérdidas en manos de los nodos extremos. Así, el tiempo mínimo empleado por el protocolo para detectar la posible pérdida de un paquete enviado desde un nodo fuente i hasta un nodo destino n, sería el tiempo total en atravesar la ruta:

$$\sum_{l=i}^{n-1} \delta_{l,l+1}$$

donde $\delta_{l,l+1}$ es el retardo del paquete al reenviarse desde el nodo l hasta el siguiente salto.

Además, en el caso óptimo de que el nodo extremo hiciera una solicitud explícita de retransmisión del paquete y sin esperar márgenes, el tiempo empleado en obtener el paquete retransmitido en el destino sería el tiempo que emplea la solicitud en llegar al origen más el tiempo empleado en retransmitirse el paquete:

$$2 \sum_{l=i}^{n-1} \delta_{l,l+1}$$

Por tanto, el tiempo total de obtención del flujo descartado en el nodo n desde el instante inicial de su transmisión sería:

$$\sum_{l=i}^{n-1} \delta_{l,l+1} + 2 \sum_{l=i}^{n-1} \delta_{l,l+1} = 3 \sum_{l=i}^{n-1} \delta_{l,l+1}$$

No obstante, en el hipotético caso de que un nodo intermedio pudiera enviar una solicitud de retransmisión local a algún nodo previo, el tiempo empleado en recuperar los datos se reduciría sustancialmente si la retransmisión de los datos perdidos se hiciera desde un nodo anterior cercano [3]. Estas retransmisiones locales o de bajo nivel podrían evitar, en parte, las retransmisiones desde el extremo origen que inician los protocolos de capas altas, resultando en un menor incremento del consumo global de ancho de banda en un dominio que ya de por sí está congestionado.

GLRP es una propuesta para la gestión cooperativa de la pérdida de paquetes para servicios con QoS [4]. Es una técnica que emplea mecanismos *Gossip* entre nodos vecinos para la distribución de la información, en contraste con los

mecanismos centralizados clásicos, en los que sólo los extremos son los responsables de diseminar información acerca de la pérdida de datos cuando los paquetes no llegan al receptor [5]. Estos algoritmos *Gossip* distribuidos presentan la ventaja de que son sencillos de implementar, haciendo que cada nodo siga reglas simples para cada evento de interés. Son también altamente tolerantes a fallos, ya que la comunicación se mantiene aunque tenga lugar un elevado nivel de pérdida de paquetes o de fallos de enlace o nodo.

La premisa subyacente a la propuesta GLRP es muy sencilla: Cuando un paquete de un flujo o servicio particular se pierde, algunos nodos intermedios del túnel LSP al que pertenece el flujo pueden seleccionar otros nodos con los que intercambiar información acerca de la pérdida de datos producida, con el objetivo de intentar retransmitir los datos descartados desde alguno de ellos [6]. GLRP se ofrece como una nueva función OAM para una gestión más rápida de las retransmisiones del tráfico perdido, para así mejorar el rendimiento de FEC prioritarios que requieran elevada fiabilidad y bajo retardo [7]. Para ello proporcionará a un número limitado de nodos intermedios del túnel LSP la habilidad de cooperar entre ellos para recuperar localmente el tráfico perdido de tales flujos prioritarios [8]. Con este propósito se almacenarán temporalmente paquetes en dichos nodos; es decir, se habilitarán más nodos desde los que recuperar paquetes perdidos o, en esencia, la retransmisión de paquetes perdidos se distribuye cooperativamente.

Estos nodos que almacenan temporalmente paquetes se denominan *nodos GLRP* y son nodos LSR a los que se les aporta la capacidad de cooperar para llevar a cabo las retransmisiones de bajo nivel de los paquetes perdidos. Incluyen una memoria que recibe el nombre de *GBuffer*, donde almacenan temporalmente paquetes y de donde los recuperan para retransmitirlos localmente. En este esquema, estos nodos GLRP intermedios que detectan la pérdida de un paquete podrán, en ese momento, enviar una solicitud de retransmisión local a alguno de los nodos GLRP anteriores. Este mensaje se denomina *GReq* e indica que el nodo que lo envía ha detectado la pérdida de un paquete y que el que lo recibe retransmitió recientemente el paquete hacia el siguiente salto sin errores. Esto se basa en la hipótesis de que un paquete descartado recientemente es también un paquete correctamente enviado por un nodo previo cercano [9]. De igual forma, el nodo receptor de un mensaje *GReq* puede, opcionalmente, devolver una respuesta indicando si tiene almacenados los datos solicitados. Esta respuesta recibe el nombre de *GAck* y puede ser positiva (*GAckOk*) o negativa (*GAckErr*). El primer caso significa que se ha encontrado el paquete solicitado en el *GBuffer* y que será reenviado inmediatamente. En el segundo caso el paquete no ha sido

encontrado, pero el nodo tiene la posibilidad de replicar el *GReq* que recibió hacia algún nodo GLRP previo para seguir intentando recuperar localmente los datos perdidos.

Observemos por tanto que cuando un nodo GLRP detecta la pérdida de un paquete dicho nodo deberá conocer el conjunto de vecinos previos con capacidad GLRP que reenviaron ese paquete. Este conjunto de nodos previos se conoce como *GPlane* de un nodo GLRP. El número de saltos necesarios para encontrar un vecino GLRP anterior desde el que se pueda llevar a cabo una recuperación local se denomina *diámetro* (d), cuyo valor, lógicamente, nunca supera la longitud del *GPlane*. A modo de ejemplo, en la Figura 3-1 se muestra el *GPlane* de un nodo X_l, el cual dispone de tres diámetros posibles para intentar recuperar localmente paquetes perdidos. Si la recuperación local se hace desde X_{l-1}, se estaría empleando un diámetro de un único salto; si se lleva a cabo desde X_{l-2} se utilizaría un diámetro de dos saltos y si se hace desde X_{l-3}, se emplearía uno de tres saltos.

Así, en el caso particular de que el nodo destino x_n fuera un nodo con capacidad GLRP, el tiempo para detectar la posible pérdida de un paquete enviado desde el nodo fuente i, seguiría siendo el tiempo total en atravesar la ruta:

$$\sum_{l=i}^{n-1} \delta_{l,l+1},$$

donde $\delta_{l,l+1}$ es el retardo del paquete al reenviarse desde el nodo l hasta el siguiente salto.

En cierto modo, el tiempo empleado en su retransmisión, suponiendo un diámetro d, sería el tiempo que emplea la solicitud en llegar al nodo "n-d" (d saltos anterior a n) más el tiempo empleado en retransmitirse el paquete desde dicho nodo hasta el destino n:

$$2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}$$

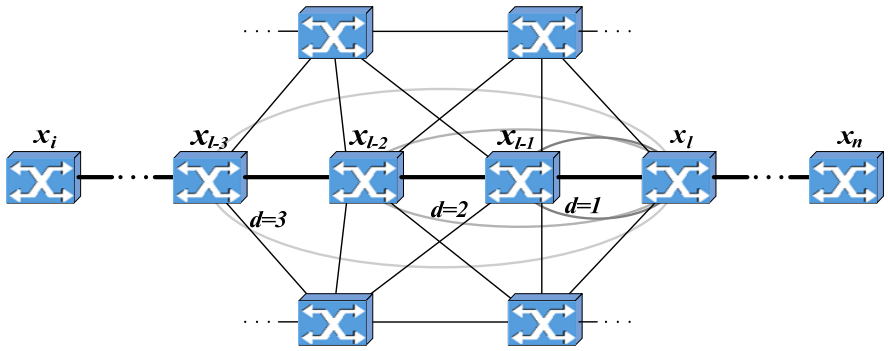

Figura 3-1. Ejemplo de *GPlane* desde un nodo X_i, con 3 posibles diámetros

Por tanto, el tiempo total de obtención del flujo descartado en el nodo n desde el instante inicial de su transmisión sería:

$$\sum_{l=i}^{n-1}\delta_{l,l+1} + 2\sum_{l=n-d}^{n-1}\delta_{l,l+1}$$

En la Figura 3-2 se muestra el funcionamiento de GLRP sobre la misma topología, en la que se reenvía un paquete desde el nodo fuente X_i hacia el destino X_n y que se descarta en un nodo intermedio X_l. Recordemos que en este caso se podrían emplear hasta 3 diámetros de *GPlane* desde X_l (d=1, d=2 y d=3) para conseguir una retransmisión local exitosa. En un primer paso, tras la detección de la pérdida del paquete, X_l enviaría una solicitud de retransmisión local (*GReq*) al primer nodo de su *GPlane* (X_{l-1}). Al recibirla, este nodo intentará localizar el paquete, para devolver una respuesta (*GAck*), incluyendo una notificación *GAckOk* para indicar que está almacenado en el *GBuffer o una notificación GAckErr* para indicar que no se ha encontrado. Si lo encuentra (esto implicaría una retransmisión local óptima, con $d = 1$ salto), copia el paquete del *GBuffer* y lo retransmite hacia su destino, terminando el proceso. Si no lo encuentra, X_{l-1} comprobaría que existen nodos GLRP previos en el *GPlane* y replicaría el mensaje *GReq* a X_{l-2}. Si el paquete no se encuentra en X_{l-2}, tampoco se conseguiría una retransmisión con $d = 2$ saltos. Nuevamente X_{l-2} comprobaría que X_{l-3} es un nodo previo del *GPlane* y le enviaría el *GReq*. En caso de no encontrarse tampoco el paquete en X_{l-3} el proceso se detiene, ya que este nodo no encontraría nodos GLRP previos en el *GPlane* de X_l, pero si lo hallara, se emplearía un diámetro de tres saltos para conseguir una retransmisión local.

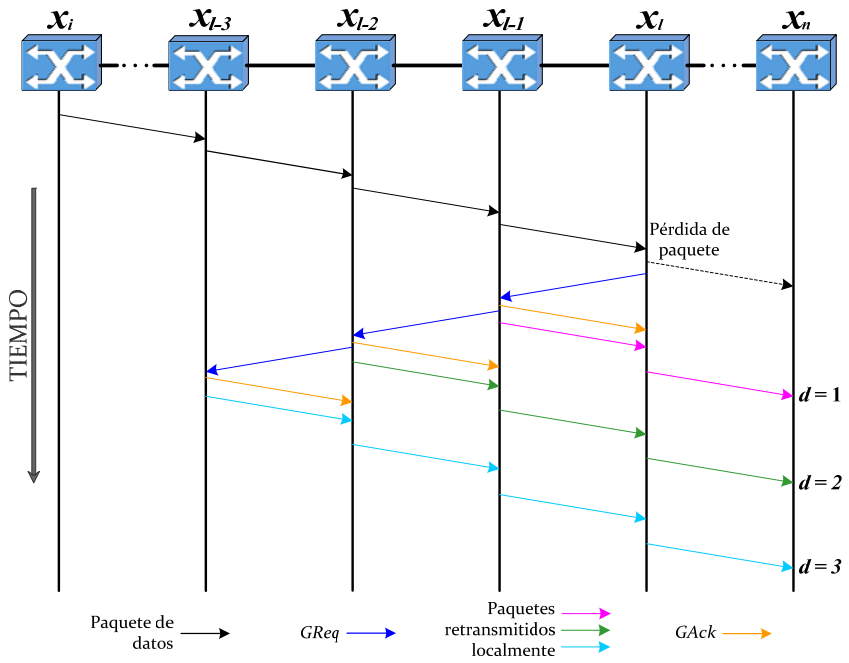

Figura 3-2. Ejemplo de recuperaciones locales desde un nodo intermedio X_l

Se debe partir de la premisa en la que cuando un nodo detecta la pérdida de datos ya debe conocer en qué nodos anteriores podrían haber sido almacenados esos datos. En cierta medida, la obtención del *GPlane* a partir de un nodo no es algo trivial. Consideremos de nuevo el dominio $G(U)$ de la topología anterior. En $G(U)$ existe un conjunto U de nodos y un flujo de paquetes $\varphi(G)=\varphi(x_i, x_n)$ que discurre a través del túnel $LSP_{0,n}$, con origen en el nodo X_i y destino en el nodo X_n y con $\{x_i, x_n\} \subset U$. En nuestro ejemplo, el flujo de paquetes que discurre a través de $LSP_{0,n}$, está superando la capacidad del nodo intermedio X_l, el cual deja de cumplir la ley clásica de Conservación del Flujo [10], es decir, la cantidad de flujo saliente de X_l sería menor que la cantidad de flujo entrante, siendo esto debido a la pérdida de uno o varios paquetes. Hasta cierto punto, a la hora de conocer cuál es el origen de los paquetes en el dominio MPLS-TP, X_l sólo conocería el puerto de entrada y la etiqueta entrante de los nuevos paquetes que llegan del FEC $\varphi(G)$, es decir, un nodo X_l sólo ve a su vecino anterior X_{l-1} como el emisor del flujo $\varphi(x_i, x_n)$ [11]. Por este motivo, antes de iniciar solicitudes *GReq*, un nodo debería conocer explícitamente el conjunto de nodos con capacidad GLRP que reenviaron dicho paquete. Con este objetivo, se proponen dos extensiones de RSVP-TE [12]. La primera permite configurar el *GPlane* en el

Plano de Control de MPLS-TP, es decir durante la señalización del LSP y la segunda se lleva a cabo en el Plano de Reenvío, habilitando la señalización de las solicitudes de retransmisión local.

3.2 GLRP en el Plano de Control MPLS

Como se ha descrito anteriormente, RSVP-TE es el encargado de señalizar los LSP en el Plano de Control de MPLS-TP [13]. Con la propuesta GLRP también se ocupará de configurar el *GPlane* en los nodos GLRP. Esta configuración consiste, simplemente en que cada nodo GLRP del LSP que se va a señalizar conozca la dirección del nodo GLRP anterior. Al mismo tiempo, se asigna al FEC lo que se denomina *GLevel* o nivel de prioridad GLRP, cuyo objetivo es el de gestionar más eficientemente las operaciones de QoS. Para GLRP el concepto de FEC está estrechamente relacionado con el *GLevel*, ya que lo utiliza para identificar y diferenciar flujos. De hecho, GLRP asigna un GLevel a cada FEC que necesite ser priorizado de manera diferente. Lleva a cabo esta clasificación principalmente en base a los requisitos de retardo y fiabilidad demandados por el flujo, asignando mejor *GLevel* a aquellos FEC con mayores requerimientos. En concreto, el *GLevel* está relacionado con el espacio que se reservará en el *GBuffer* de los nodos GLRP del *GPlane*. Es decir, la asignación de un mejor *GLevel* a un FEC implica que se reservará más espacio en los *GBuffer* para los paquetes de ese FEC. Este hecho aumenta la probabilidad de recuperar localmente los paquetes perdidos empleando diámetros menores disminuyendo, por tanto, el tiempo empleado en retransmitir los datos perdidos.

En este contexto, GLRP extiende el protocolo RSVP-TE para definir el *GPlane* como un subconjunto de nodos de un LSP prioritario. Recordemos que en el Plano de Control de MPLS-TP, cuando un LER de entrada del dominio recibe un mensaje solicitando la creación de un nuevo LSP, se inserta una nueva entrada en la LFIB con información sobre cómo se reenviarán los paquetes a través del LSP que se está señalizando. Esta es la información que emplearán luego los LSR en el Plano de Reenvío cuando reciban paquetes etiquetados y deban hacer la conmutación de etiquetas para reenviarlos hacia el siguiente salto. Si se trata de un LSP para un FEC prioritario, los nodos GLRP también insertarán el *GLevel* y la dirección del nodo GLRP previo (*GPHOP*) en una tabla denominada *GTable*. Así se estará señalizando un "*GPlane* orientado a conexión", el cual permitirá que cuando uno de los nodos GLRP detecte la pérdida de un paquete de este FEC, dicho nodo podrá consultar la *GTable* para

obtener toda la información que necesita para iniciar una solicitud de retransmisión local.

La Tabla 3-1 muestra un ejemplo de la *GTable* de un nodo que reenvía paquetes de hasta 4 FEC diferentes. Cada fila contiene una primera columna que identifica al FEC, mediante la combinación de etiqueta entrante y saliente. La segunda columna incluye el GLevel asignado al FEC y, finalmente, la tercera columna almacena la dirección del anterior nodo GLRP al que enviar las solicitudes de retransmisión local en caso de pérdidas de paquetes. Observemos, por ejemplo que los paquetes del FEC 36/68 y los del 108/44 se reenvían a través del mismo *GPHOP*, cuya dirección es x.x.160.17. Sin embargo, se ha reservado más espacio en el *GBuffer* para el primero, por su mejor *GLevel*, por lo que se deduce que habrá más probabilidades de recuperar localmente más rápidamente los posibles paquetes perdidos.

En particular se extienden los mensaje *Path* y *Resv* de RSVP-TE. En el primero se incluye un objeto simple denominado *GPath* y en el segundo un objeto llamado *GResv*. *GPath* se define como una solicitud de creación de un *GPlane* e incluye información GLRP, como el *GLevel* del FEC y el nodo previo en el *GPlane*. El nodo tomará estos datos para solicitar la inserción de una nueva fila en su *GTable* para el FEC que está señalizando RSVP-TE. El objeto *GResv* incluido en el mensaje *Resv* se empleará como confirmación para dicha solicitud. La Figura 3-3 describe el comportamiento de un nodo GLRP en el Plano de Control MPLS. Se parte de un nodo GLRP que está recibiendo mensajes RSVP-TE para la señalización de un nuevo LSP en el dominio.

Tabla 3-1. Ejemplo de *GTable* para 4 FEC prioritarios

Etiqueta entrante / Etiqueta saliente	**GLevel**	**Dirección del *GPHOP***
4/32	11	x.x.160.12
36/68	1	x.x.160.17
108/44	18	x.x.160.17
74/60	4	x.x.160.35

Si se recibe el mensaje *GPath* opcionalmente se puede llevar a cabo un mecanismo común de control de admisión. Es decir, se comprobará si el *GLevel* solicitado es factible en función de los *GPlane* ya configurados y del espacio que se puede reservar aún en el *GBuffer*. Si la nueva reserva pudiera afectar a los *GPlane* ya configurados, opcionalmente se podría cancelar la señalización del LSP mediante el envío de un mensaje *PathErr* hacia el LER de entrada. En otro caso, se aceptaría la solicitud y se añadiría una nueva fila en la *GTable*, incluyendo el identificador de FEC, el *GLevel* solicitado y la dirección del nodo GLRP previo, si lo hubiera. Esta fila queda pendiente de confirmar, lo que se hará al recibir el correspondiente mensaje *GResv* enviado desde el siguiente salto del LSP, siguiendo la filosofía común de señalización de LSP mediante la combinación de mensajes *Path*/*Resv*. Tras añadir la fila en estado pendiente en la *GTable* se grabaría la dirección del nodo actual en el mensaje *GPath* y se reenviaría hacia el siguiente salto, para que un nodo GLRP posterior pueda conocer que el nodo actual es su *GPHOP*.

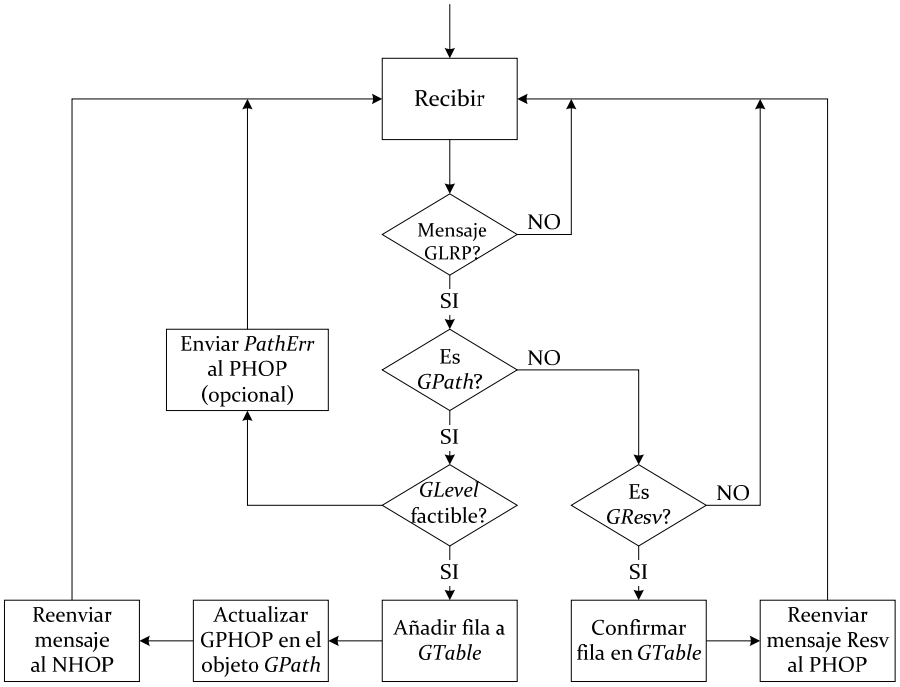

Figura 3-3. Plano de Control de GLRP

3.3 GLRP en el Plano de Reenvío MPLS

En el Plano de Reenvío un nodo GLRP conmuta etiquetas y reenvía paquetes de datos hacia el siguiente salto, al igual que un LSR común [14]. Si bien, se pueden producir dos eventos que provocan un cambio de estado en el nodo. El primero es la detección de una pérdida de paquete de un FEC prioritario. En este caso, el nodo GLRP toma el FEC y consulta en la *GTable* la dirección del salto previo del *GPlane* y cambia su estado a *Solicitud de recuperación local*, para proceder a enviar un mensaje *GReq* a su *GPHOP*. Cuando se reciba un mensaje de confirmación (*GAck*) para esa petición, cambiará su estado de nuevo al estado inicial (ver Figura 3-4). El otro evento que produce un cambio de estado es la recepción de un mensaje *GReq* enviado desde algún nodo siguiente del *GPlane*. En esta ocasión, el nodo cambia su estado a *Acceso al GBuffer*, para buscar el paquete solicitado, según la información recibida en la solicitud *GReq*. Si se localiza el paquete se envía un mensaje *GAck* positivo en respuesta al *GReq*, para indicar que el paquete solicitado se ha encontrado y que será retransmitido localmente. Después cambia al estado *Retransmisión local* para copiar el paquete del *GBuffer*, reenviarlo y pasar de nuevo al estado inicial. En caso de no haber localizado el paquete solicitado, devuelve un mensaje *GAck* negativo y, si existe algún *GPHOP* cambia al estado *Solicitud de recuperación local*, para replicar el mensaje *GReq* a dicho nodo previo del *GPlane*.

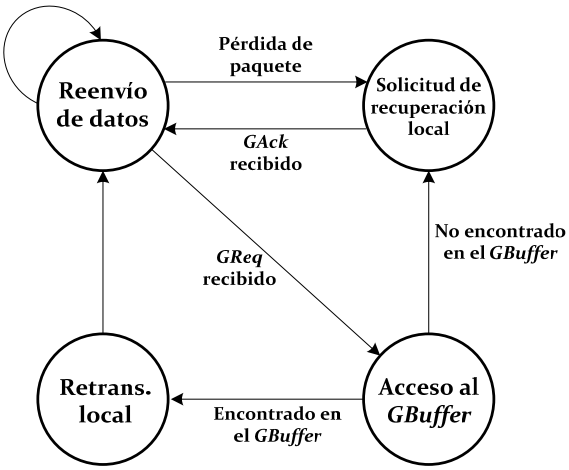

Figura 3-4. Diagrama de estados de un nodo GLRP en el Plano de Reenvío

Al mismo tiempo, para optimizar el acceso al *GBuffer*, los nodos GLRP emplean la denominada tabla *GIndex*, cuyo objetivo es el de mejorar la eficiencia de la búsqueda de paquetes cuando se reciben solicitudes de retransmisión local. Esta tabla habilita el acceso aleatorio o directo al *GBuffer* ya que, en lugar de recorrer todo el buffer intentando localizar el paquete solicitado, la tabla *GIndex* indica la posición en la que está almacenado un paquete particular en el *GBuffer*. Para ello la tabla asocia un valor numérico o índice a cada paquete guardado, que será el puntero a la posición del *GBuffer* en la que está almacenado el paquete. El objetivo es que los paquetes se puedan recuperar eficientemente, independientemente del tamaño del *GBuffer*, ya que no hay que recorrer todo el buffer para localizar cada paquete solicitado. Además, si el paquete no está almacenado en el *GBuffer* el índice también lo indica, evitando la necesidad de recorrerlo para hacer la comprobación. Para todo este cometido se emplea una función *Perfect Hash*, la cual es una función *hashing* simple que asocia los diferentes paquetes recogidos en el *GBuffer* a un conjunto de números enteros. *Perfect Hash* es una función muy eficiente, con búsquedas a velocidad constante y que permite gestionar hasta un millón de índices en pocos segundos de procesamiento, si fuera necesario [15].

En la Figura 3-5 se detalla el comportamiento de un nodo GLRP en el Plano de Reenvío. Cuando un nodo recibe un paquete primero comprueba si pertenece a un FEC prioritario. En tal caso, y si se descarta, se accederá a la *GTable* para comprobar el *GPlane* del nodo. Si existe algún nodo GLRP previo se le enviaría una solicitud *GReq*, pero si no encuentra otro nodo GLRP simplemente no entra en acción y queda en manos de las capas altas la retransmisión de los datos perdidos desde el extremo origen. Si el paquete no fue descartado, se debe comprobar si se trata de un paquete de datos o de un mensaje GLRP. En el primer caso, al mismo tiempo que se reenvía el paquete al siguiente salto del LSP, también se almacena en el *GBuffer* y se actualiza el *GIndex*. Si en cambio se trataba de un mensaje GLRP, esto significa que un nodo posterior del *GPlane* detectó la pérdida de un paquete y está solicitando al nodo actual la retransmisión local del mismo. Para ello accede a *GIndex* para comprobar si el paquete está almacenado, en cuyo caso lo recuperaría del *GBuffer* y lo retransmitiría hacia el siguiente nodo. Si *GIndex* indica que el paquete solicitado no se encuentra en *GBuffer*, habría que comprobar en *GTable* si existe un *GPHOP*. Si es así se replicaría el *GReq* recibido hacia dicho nodo, pero en caso contrario GLRP se detendría y la retrasmisión debería hacerse extremo a extremo.

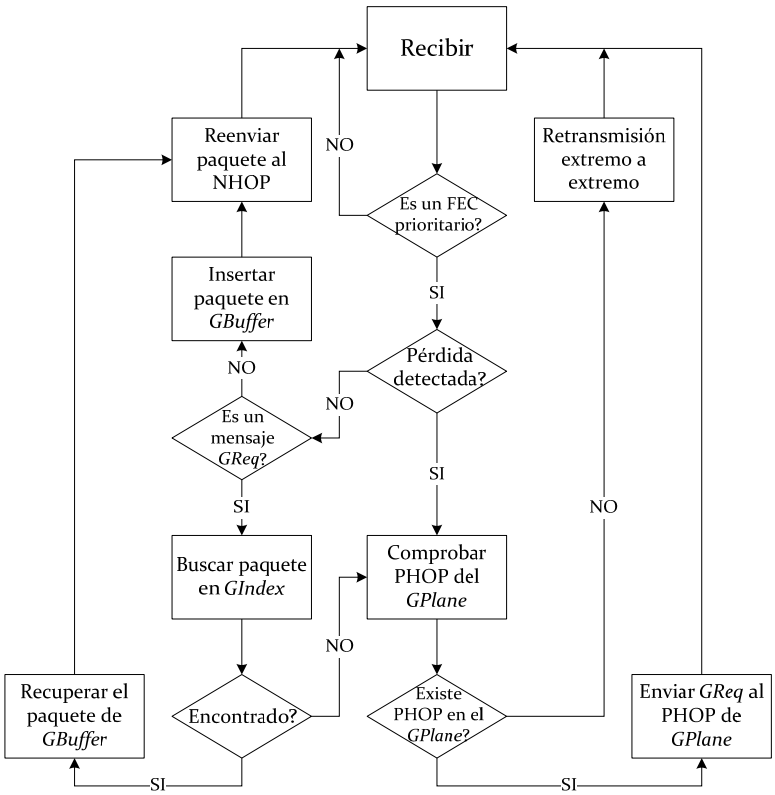

Figura 3-5. Plano de Reenvío de GLRP

Periódicamente se pueden borrar las entradas más antiguas de la tabla *GIndex*, con el objetivo de evitar que esta tabla crezca indefinidamente. Para ello, cada entrada de *GIndex* tiene asociada una marca de tiempo o *timestamp*, con el valor del instante en que se insertó. Las entradas que hayan superado el tiempo máximo de espera por una hipotética solicitud de retransmisión local del paquete asociado se pueden borrar. Es decir, si una entrada de *GIndex* no ha sido accedida por un tiempo superior a dicho tiempo de espera, puede borrarse. Este periodo de espera depende de RTT_d, que es el *Round Trip Time* entre el nodo de recuperación y el nodo que detecta la pérdida, separados por un diámetro igual a d saltos. El valor de RTT_d también se emplea para elegir qué nodo debe ser reemplazado por un nuevo paquete de datos entrante en el *GBuffer*. En este caso, cuando un paquete ha sido sobrescrito, las futuras solicitudes de retransmisión local para ese paquete se reenviarán hacia nodos GLRP anteriores. Esta situación

es más probable si se ha asignado un nivel GLRP bajo al FEC y, por tanto, una menor reserva de espacio en *GBuffer*. De hecho, un paquete sólo se borra de *GBuffer* cuando es sobrescrito por un nuevo paquete de datos entrante. Por otra parte, si un nodo GLRP cercano ha borrado un paquete, esto no implica que nodos anteriores más lejanos también lo hayan hecho. Por ejemplo, si un nodo cercano está sometido a un mayor tráfico cruzado que otros nodos previos más lejanos, el nodo cercano borrará un paquete particular antes que los nodos más lejanos, los cuales podrían almacenar el paquete por más tiempo. Por tanto, a pesar de que un paquete haya sido sobrescrito, la entrada asociada en *GIndex* se mantiene, ya que ésta sólo se borra cuando no exista posibilidad de recibir un *GReq* para ese paquete. Así, en caso de recibir alguna solicitud de retransmisión local, se pueda seguir contestando sin acceder al *GBuffer* que el paquete ya no está almacenado. El *timestamp* asociado a esa entrada se empleará para calcular el tiempo máximo de espera durante el que la entrada debe estar activa y el objetivo sólo es evitar que la tabla *GIndex* crezca indefinidamente.

3.4 Formato de los objetos *GPath*, *GResv*, *GReq* y *GAck*

El formato de mensaje RSVP-TE es muy sencillo. Está compuesto por una cabecera común (ver Figura 3-6), cuyos campos se describen en la

Tabla 3-2. Esta cabecera común va seguida de un número variable de objetos, lo que depende del mensaje RSVP-TE en sí. Además, todos tienen una estructura TLV (Tipo-Longitud-Valor), como se muestra en la Figura 3-7. Es decir, cualquier objeto RSVP-TE incluye una cabecera propia en la que se indica el *Tipo*, para identificar el objeto, su *Longitud*, ya que pueden incluir un número variable de campos y *Valor*, con el contenido particular del objeto, como se describe en la Tabla 3-3.

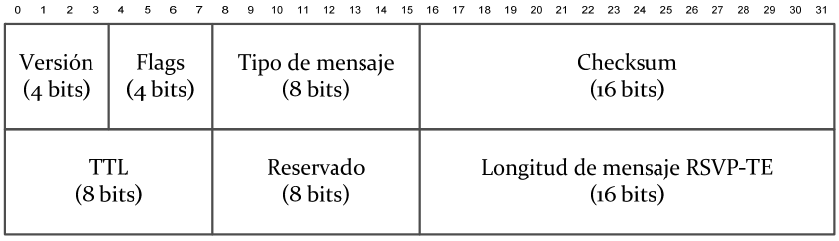

Figura 3-6. Formato de cabecera común RSVP-TE

Tabla 3-2. Campos incluidos en la cabecera común RSVP-TE.

Campo	**Descripción**
Versión	Versión del protocolo RSVP-TE.
Flags	No definidas.
Tipo de mensaje	1 = *Path* 2 = *Resv* 3 = *PathErr* 4 = *ResvErr* 5 = *PathTear* 6 = *ResvTear* 20 = *Hello*
Checksum	Control de errores en el mensaje RSVP-TE.
TTL	Es el TTL del paquete IP con el que se transmite el mensaje RSVP-TE.
Reservado	Para usos futuros.
Longitud	Longitud del mensaje, medido en octetos, incluyendo la cabecera común, por lo que, como mínimo debe tener valor 8.

Los mensajes *GPath*, *GResv*, *GReq* y *GAck*, propios de GLRP, se implementan como nuevos objetos TLV sencillos, que extienden el protocolo RSVP-TE [16]. Los dos primeros extienden los mensajes *Path* y *Resv*, respectivamente, como se ilustra en la Tabla 3-4 y en la Figura 3-8 permiten la configuración del *GPlane* como un subconjunto de nodos del LSP.

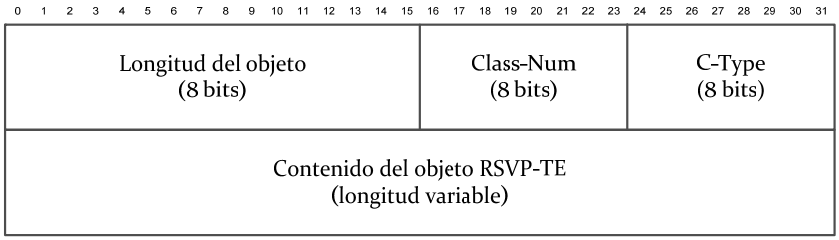

Figura 3-7. Formato de objeto RSVP-TE

Tabla 3-3. Descripción de los campos de la cabecera de objeto RSVP-TE

Campo	**Descripción**
Longitud	Longitud del objeto en octetos, incluyendo la cabecera del objeto.
Class-Num	Clase del objeto.
C-Type	Tipo de la clase (es un identificador único dentro de la clase).
Contenido	El valor del objeto. Incluye un número variable de campos.

Tabla 3-4. Descripción de los mensajes *Path* y *Resv*, incluyendo objetos GLRP

	Objeto	**Descripción**
Objetos incluidos en ambos mensajes	Session	Identifica el túnel que se está señalizando, incluyendo parámetros como las direcciónes del router de entrada y del de salida.
	RSVP-Hop	Contiene la dirección de la interfaz desde la que se envió este mensaje.
	Time_Values	Intervalo de tiempo (ms) entre mensajes *Path* o *Resv* consecutivos.
	Record_Route (RRO)	Almacena la dirección y etiqueta de cada salto del túnel LSP que se está señalizando.
Objetos incluidos en el mensaje *Path*	Explicit_Route (ERO)	Conjunto de routers a través de los cuales debe señalizarse el túnel.
	Label_Request	Etiqueta que se sugiere al siguiente salto. Debe confirmarse en el mensaje *Resv* correspondiente.
	Sessión_Attribute	Incluye atributos como la prioridad de señalización o de mantenimiento, protección local o almacenado de etiquetas.
	Sender_Template	Especifica la dirección del router de entrada y el ID del túnel LSP.
	Sender_Tspec	Especifica la demanda de recursos para el túnel. Incluye parámetros como velocidad media, tamaño del *Token Bucket*, velocidad máxima, tamaño máximo de paquete, etc.
	Adspec	Contiene información acerca de la ruta que toma el LSP que se está señalizando.
	GPath	Objeto específico de GLRP. Es una solicitud de creación de un *GPlane*. Incluye atributos como ID de flujo, solicitud de *GLevel* para ese flujo y dirección del nodo previo del *GPlane*.
Objetos incluidos en el mensaje *Resv*	Style	Especifica el tipo de reserva de recursos para el túnel LSP.
	Flowspec	Similar a *Sender_Tspec* del mensaje *Path*.
	Filter_Spec	Similar a *Sender_Template* del mensaje *Path*.
	Label	Etiqueta de 20 bits que el nodo previo debería emplear en los paquetes de datos para el túnel LSP que se está señalizando.
	GResv	Objeto específico de GLRP. Es la confirmación a una solicitud de creación de *GPlane*. Incluye el *GLevel* asignado a un flujo de datos particular al emplear el *GPlane* que se está señalizando.

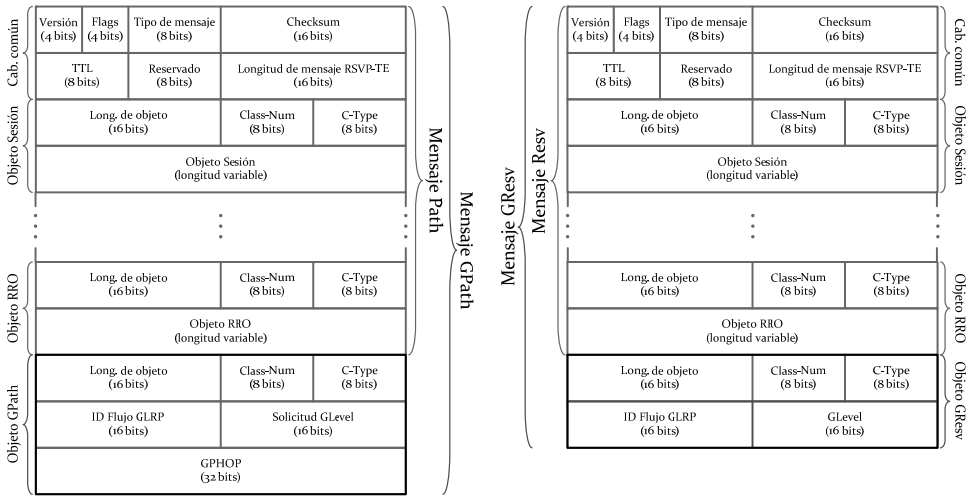

Figura 3-8. Formato de los mensajes *GPath* y *GResv*, como extensiones de los mensajes *Path* y *Resv*, respectivamente

Los mensajes *GReq* y *GAck* extienden el mensaje *Hello*, como se muestra en la Tabla 3-5 y en la Figura 3-9 y permiten la señalización de solicitudes de retransmisión local en el Plano de Reenvío.

Tabla 3-5. Descripción del mensaje *Hello*, incluyendo objetos GLRP

Objeto	**Descripción**
Hello	Permite detectar vecinos con los que se pierde la comunicación mediante un mecanismo de detección punto a punto. Incluye dos contadores (*Source Instance* y *Destination Instance*) para comprobar el estado del vecino RSVP-TE. Sigue la filosofía de funcionamiento clásica de un *keepalive*.
GReq	Objeto específico de GLRP. Es una solicitud de retransmisión local. Contiene el identificador de flujo prioritario para GLRP del que se solicita la retransmisión local de datos.
GAck	Objeto específico de GLRP. Es la confirmación de una solicitud de retransmisión local. Indica si los datos solicitados han sido encontrados en el *GBuffer* o si se va a replicar la solicitud al *GPHOP*, si no se van a poder recuperar localmente, si se ha producido algún error, etc.

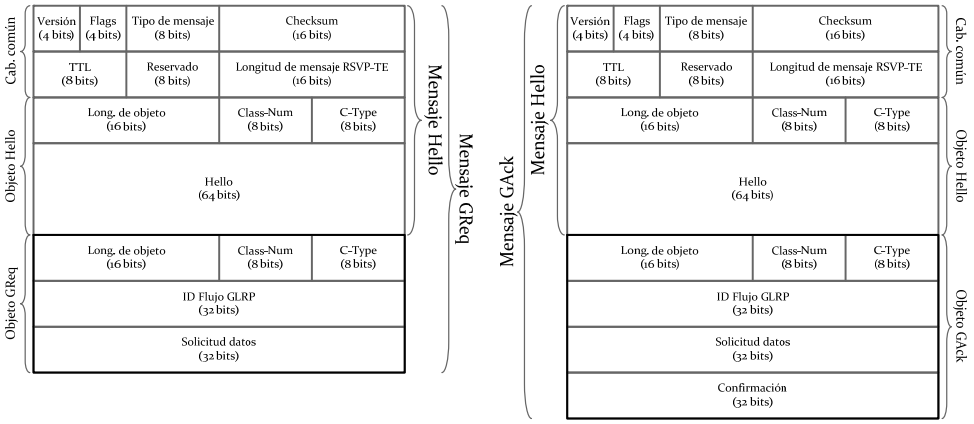

Figura 3-9. Formato de los mensajes *GReq* y *GAck*, como extensiones de los mensajes *Hello Request* y *Hello Ack*, respectivamente

3.5 Arquitectura del nodo GLRP

Una de las metas de las técnicas OAM de MPLS-TP es proporcionar las herramientas necesarias para monitorizar y gestionar la red teniendo en cuenta los parámetros empleados por las tecnologías clásicas de transporte. En este sentido GLRP es una técnica de mantenimiento OAM para MPLS-TP y tiene el objetivo de minimizar el número de paquetes perdidos que se deberían recuperar extremo a extremo, evitando así la degradación del servicio en cuanto a fiabilidad y *delay* de entrega de paquetes en aplicaciones o flujos privilegiados. Siguiendo la filosofía de OAM, GLRP se ha diseñado para señalizarse en banda, es decir, a través de la misma ruta o LSP que emplean los paquetes de datos [17].

Primero, la señalización del *GPlane* se lleva a cabo en el Plano de Control, mediante el protocolo RSVP-TE, durante la fase de creación del túnel LSP que emplearán los paquetes de datos. Después, en el Plano de Reenvío los mensajes GLRP se señalizan también en banda, junto a los paquetes de datos u otros mensajes OAM y utilizando el LSP bidireccional que recomienda crear MPLS-TP [18].

En la Figura 3-10 se muestra la arquitectura de un nodo con capacidad GLRP. Se ha extendido la arquitectura de nodo MPLS, integrando la funcionalidad GLRP y mostrando también los mensajes GLRP de los Planos de Control y de Reenvío.

En el Plano de Control el nodo MPLS lleva a cabo dos tareas claramente diferenciadas. Por un lado, debe intercambiar información relativa a la distribución de etiquetas con sus vecinos, con el objetivo de señalizar túneles LSP. Esta información la emplea para ir rellenando la tabla LFIB. Por otro lado, también intercambia información de encaminamiento propia de Capa 3, ya que es un nodo apto también para el reenvío de paquetes no etiquetados, por lo que debería tomar decisiones sobre el mejor siguiente salto en función de la dirección del nodo destino transportada con los paquetes. Esta información la utiliza para rellenar la tabla de encaminamiento de Capa 3. Al mismo tiempo hemos dotado al nodo de capacidad GLRP en este plano, para permitir el intercambio de información relacionada con la creación del *GPlane* (mensajes *GPath* y *GResv*) del túnel LSP que se esté señalizando a través de ese nodo.

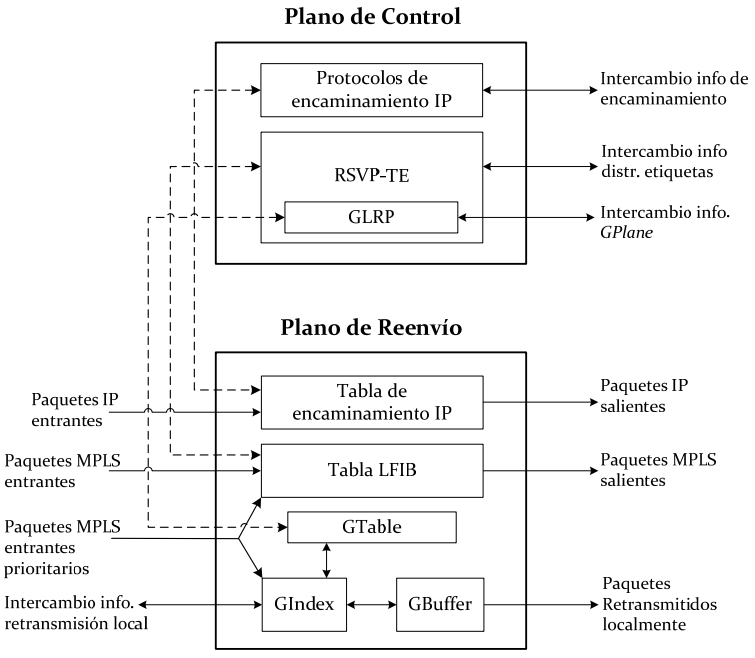

Figura 3-10. Arquitectura de nodo MPLS-TP con capacidad GLRP

En el Plano de Reenvío, por un lado, el nodo puede recibir paquetes no etiquetados, para los que necesitará acceder a la tabla de encaminamiento de Capa 3 y tomar en ese momento la decisión de cuál es el mejor siguiente salto para cada paquete en función de su dirección destino. Por otro lado, también puede recibir paquetes etiquetados. En este caso debe acceder a la tabla LFIB, rellenada en el Plano de Control, para decidir cómo se debe hacer la conmutación de etiquetas y cómo se debe reenviar el paquete hacia el siguiente salto del túnel LSP. Una entidad de clasificación de paquetes puede tomar decisiones sobre el tipo o servicio al que pertenecen, y así llevar a cabo políticas de priorización de paquetes [19]. El clasificador es útil a GLRP también para decidir si los paquetes se deben almacenar en el *GBuffer* y el espacio que les corresponde, lo que viene determinado por el valor de *GLevel*, como se ha comentado anteriormente. Así mismo el nodo puede recibir paquetes de solicitud de retransmisión local. En este caso comprueba en *GIndex* si los datos solicitados se encuentran almacenados en *GBuffer*. Si es así accede directamente a la posición de los datos, los recupera y los retransmite localmente hacia el *NHOP*. Si no están almacenados, accede a la *GTable* para comprobar si existe un nodo *GPHOP* al que replicar la solicitud *GReq*.

En la Figura 3-11 se muestra una topología sencilla en la que el clasificador tradicional MPLS recibe paquetes etiquetados, extrae la cabecera MPLS, comprueba el valor de la etiqueta y categoriza el paquete a partir de la información de QoS inferida desde el valor de etiqueta. El paquete, finalmente pasará a la cola de la interfaz de salida según lo indicado por la LFIB para esa etiqueta entrante; en la cola el paquete será tratado con una prioridad que dependerá de la categoría que le haya asignado el clasificador. Sin embargo, con la propuesta GLRP, si se detecta la pérdida de un paquete prioritario para GLRP (con elevado *GLevel*), se enviará un mensaje *GReq* al nodo previo del *GPlane*, el cual accederá al *GIndex* para detectar si el paquete está almacenado o no. En el primer caso, devolverá un mensaje *GAck* positivo para luego retransmitir localmente el paquete. En el segundo caso restituirá un *GAck* negativo y replicará el mensaje *GReq* a su *GPHOP*, si existiera.

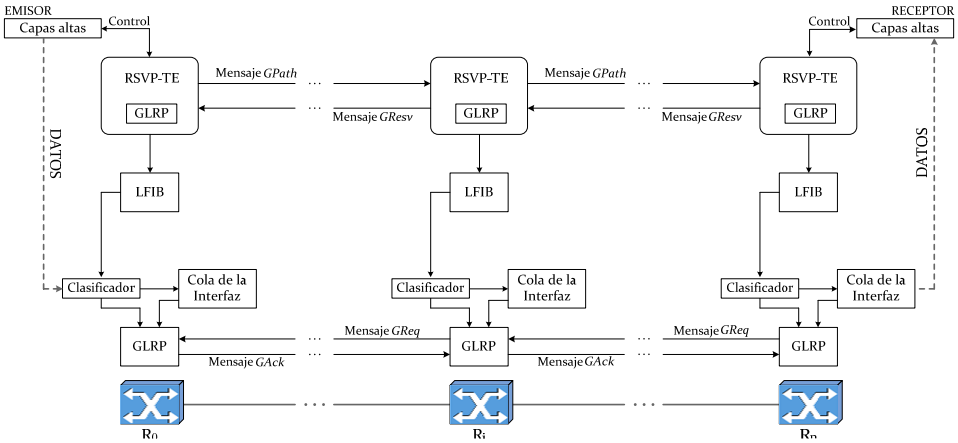

Figura 3-11. Intercambio de mensajes GLRP entre nodos de un *GPlane*

3.6 GLRP para servicios no orientados a conexión

La variante CL-GLRP (*ConnectionLess GLRP*) se ofrece como servicio especializado para flujos de paquetes no etiquetados. Para ello, es necesario que los nodos GLRP por los que pasen los paquetes inserten su dirección IP en la cabecera del paquete, con el objetivo de que el nodo que solicite una retransmisión local pueda conocer el *GPlane* por el que ha pasado el paquete. Por tanto, atendiendo al formato de cabecera, se plantean dos posibilidades, una propuesta para IPv4 y otra para IPv6.

Si se trata de flujos IPv4, se puede emplear el campo *Opciones* de la cabecera, mostrada en la Figura 3-12 para transportar, tanto el conjunto de nodos GLRP por los que ha pasado el paquete, como el *GLevel* asignado a dicho flujo de paquetes. Debido al propio tamaño máximo de este campo, el diámetro máximo del *GPlane* quedará limitado a 8 saltos.

La trama IPv4 está constituida por una cabecera obligatoria, el campo *Opciones*, que es opcional y el campo de datos. En cambio la estructura de IPv6 está formada por una cabecera obligatoria de 40 octetos, seguida de una o varias cabeceras de extensión opcionales (ver Figura 3-13).

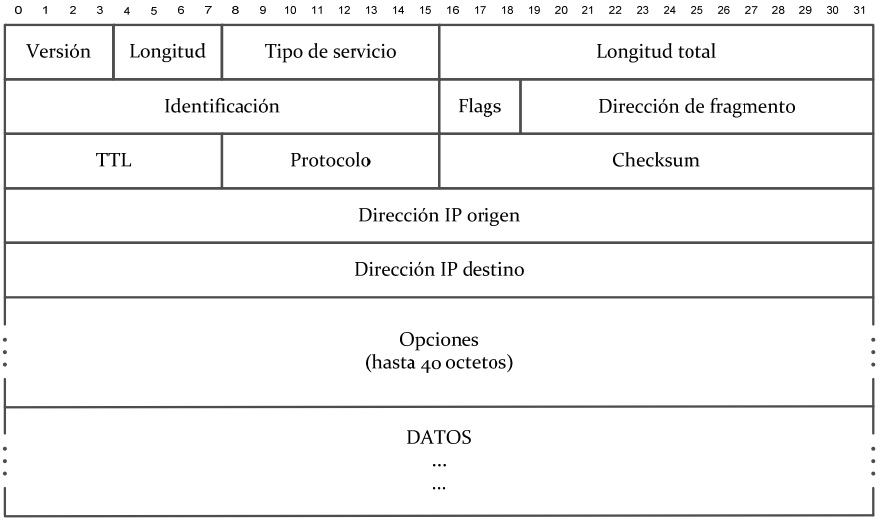

Figura 3-12. Formato de cabecera IPv4

Aunque la cabecera IPv6 es más grande que la parte obligatoria de la cabecera IPv4, la primera contiene menos campos. De esta forma, los routers IPv6 tienen que hacer un menor procesamiento por cada paquete, agilizando así el encaminamiento [20]. En la Tabla 3-6 se muestra una comparativa del uso de cada campo en ambas versiones.

En consecuencia, CL-GLRP debe emplear una cabecera de extensión opcional para señalizar el *GPlane*. Las cabeceras de extensión opcionales definidas se recogen en la Tabla 3-7.

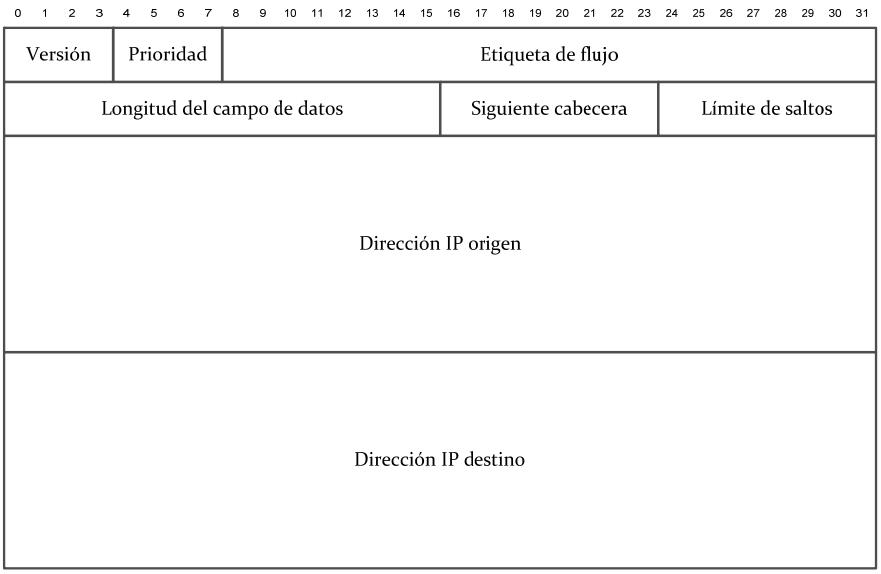

Figura 3-13. Formato de cabecera IPv6

Tabla 3-6. Uso de cada campo en las cabeceras IPv4 e IPv6

Campo de cabecera IPv4	**Campo equivalente en IPv6**
Longitud de cabecera	Campo eliminado, la cabecera IPv6 tiene tamaño fijo.
Tipo de Servicio	Convertido en los campos *Prioridad* y *Etiqueta de Flujo*.
Longitud Total	Transformado en el campo *Longitud del campo de datos*.
Identificación / Desplazamiento	Incluidos en la *Cabecera de Fragmentación*.
TTL	Campo *Límite de saltos*.
Protocolo	Campo *Siguiente Cabecera*.
Checksum	No se comprueba errores en IPv6 porque ya lo hacen los protocolos de capas superiores
Opciones	En las cabeceras de extensión opcionales

Tabla 3-7. Descripción de las cabeceras de extensión opcionales de IPv6

Cabecera	**Descripción**
Salto a salto	Transporta información opcional que puede examinar cada router de la ruta.
Encaminamiento	Contiene una lista de nodos intermedios por los que debe pasar el paquete en su camino hacia el destino.
Fragmentación	En IPv6 la fragmentación sólo la puede llevar a cabo el nodo origen. El nodo que ejecuta el algoritmo para obtener la mejor ruta hacia un destino podrá también conocer la MTU permitida de cada red por la que pase el paquete. Así el nodo origen fragmentará el paquete, según se requiera, para cada dirección de destino dada.
Autenticación y encapsulamiento	Cabecera relacionada con la seguridad nativa de IPv6.
Opciones para el destino	Equivalente a la cabecera de opciones salto a salto, pero transportando opciones que sólo las examinará el nodo destino.

La cabecera *opciones salto a salto*, por consiguiente, se presenta como la más adecuada para transportar la información GLRP, que debe ser examinada por nodos intermedios [21]. Esta cabecera consta de los campos *Cabecera siguiente*, *Longitud de cabecera* y *Opciones* (ver Figura 3-14). El campo *Opciones*, de longitud variable, es el que contiene la opción en cuestión. Ésta se expresa mediante el campo *Tipo de opción* (1 byte) que identifica la opción, *Longitud* (1 byte), que especifica el número de bytes que ocupa la opción y los *Datos de la opción*. Los 5 bits menos significativos del campo *tipo de opción* identifican la opción. Los 2 bits más significativos indican la acción que tiene que realizar un nodo que no reconozca dicha opción:

- 00: Ignorar la opción y continuar procesando.
- 01: Descartar el paquete.
- 10: Descartar y enviar mensaje ICMP al nodo origen para notificar el problema.
- 11: Descartar y enviar notificación ICMP de problema al nodo origen si la dirección destino no es de multidifusión.

Finalmente, el tercer bit indica si el campo de datos de la opción no cambia (valor=0) o si puede cambiar (valor=1) desde el origen al destino.

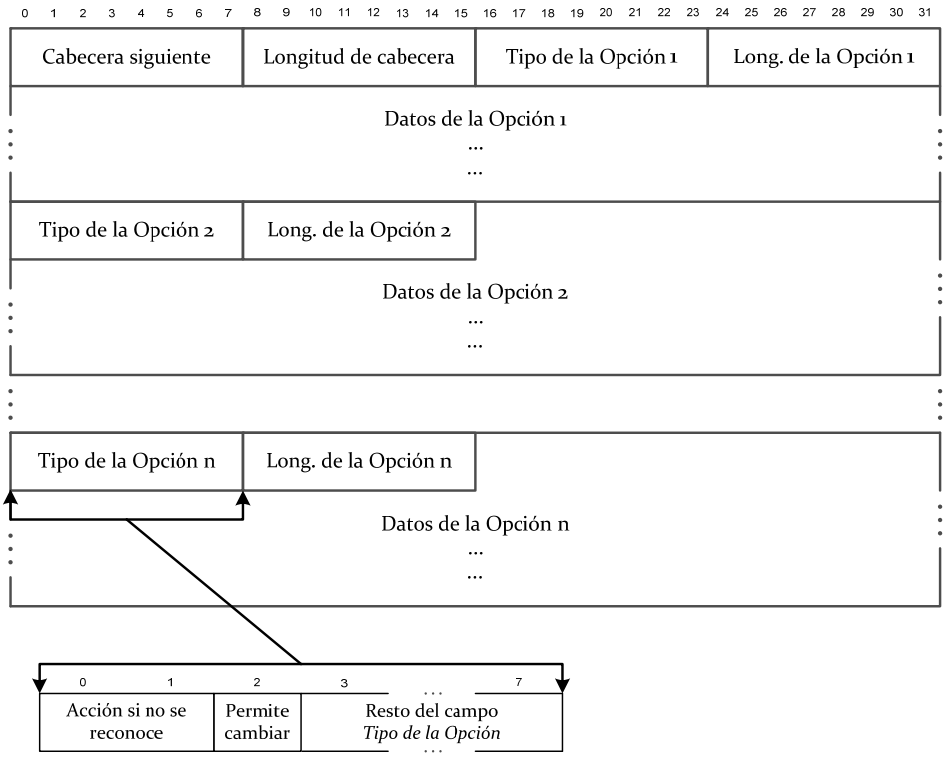

Figura 3-14. Cabecera de Opciones Salto a Salto de IPv6.

Algunas opciones se han definido ya para la cabecera Sato a salto, como *Relleno1* ó *RellenoN*, para insertar 1 ó n bytes de relleno, respectivamente, para asegurar que la longitud es múltiplo de 8 octetos; *Jumbogramas*, para el soporte de paquetes con payload mayor de 65.535 bytes (el campo longitud de carga útil en la cabecera fija IPv6 es de sólo 16 bits); *Alertas al nodo*, para informar a nodos particulares de que el paquete es de interés para él o que contiene información de control. Esta última opción es especialmente útil para proporcionar apoyo a protocolos como RSVP, que generan paquetes que deben ser analizados por nodos intermedios para el control de tráfico [22]. Esta opción se encargará de avisar al router cuando se requiera el análisis detallado de la cabecera. Así el resto del tiempo el nodo no se dedicará a dicha labor, optimizándose el proceso de reenvío [23]. En lo que se refiere a GLRP, la información de señalización se incluye como una opción nueva en la cabecera de opciones salto a salto, estableciéndose los 3 bits de mayor peso a 0, para que los nodos sin capacidad GLRP simplemente la ignoren (ver Figura 3-15).

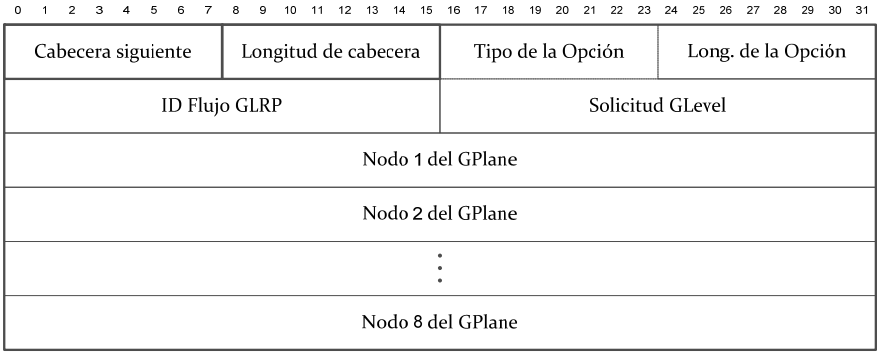

Figura 3-15. Opción GLRP en la cabecera de Opciones Salto a Salto de IPv6.

3.7 Recuperación local de ráfagas de paquetes

Adicionalmente, GLRP soporta la recuperación de bloques de paquetes, con el objetivo de reducir el número de solicitudes de retransmisión local. Esto es especialmente útil en dominios congestionados en los que, cuando se pierde un paquete, existe gran probabilidad de que otros paquetes se descarten también en un periodo corto de tiempo, o incluso consecutivamente [24]. Así, el nodo que pierde paquetes puede solicitar en un solo mensaje la retransmisión local de varios paquetes simultáneamente, en lugar de enviar una solicitud por cada paquete a retransmitir. Con esto se persigue el objetivo de optimizar la señalización en entornos propensos a ráfagas de pérdidas [25]. En la

Figura 3-16 se puede observar el formato de mensaje *GReq* para solicitar múltiples paquetes perdidos.

Su funcionamiento varía muy poco con respecto a lo ya estudiado: Cuando un nodo GLRP del *GPlane* recibe una solicitud de retransmisión de un bloque de paquetes desde algún nodo siguiente, busca cada paquete en *GIndex*. Los que encuentre, los recupera del *GBuffer* y los retransmite hacia el siguiente salto. Estos paquetes encontrados son borrados de la solicitud del bloque y el resto continuará en el mensaje de solicitud de bloque que se replicará hacia el anterior nodo del *GPlane*. Este funcionamiento se muestra en la Figura 3-17.

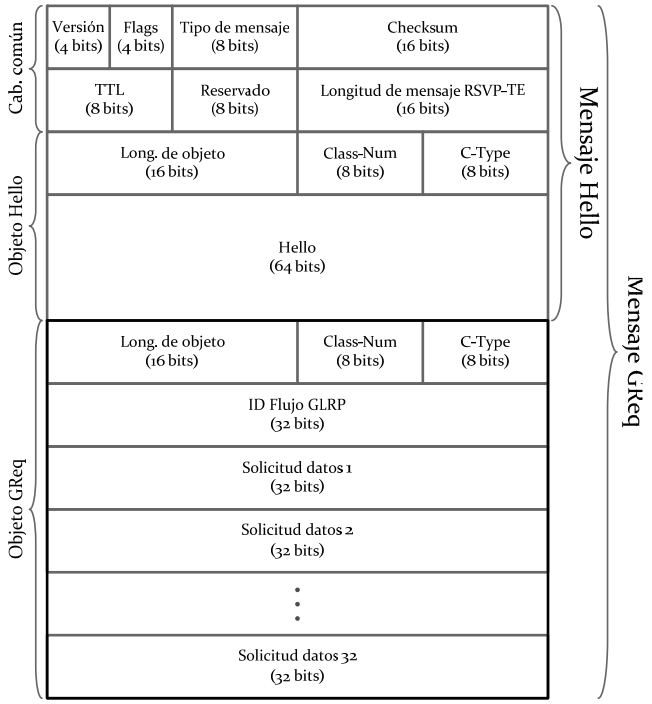

Figura 3-16. Mensaje GReq para la solicitud de un bloque de paquetes

Por otro lado, en el mensaje *GAck* se codifica el campo *Confirmación* de forma que en los 4 bytes se pueda informar sobre todos los paquetes del bloque. Cada bit hace referencia a un paquete solicitado. Si un bit tiene valor 1, esto indica que el paquete solicitado se ha encontrado y pronto se retransmitirá localmente y si tiene valor 0 indica que no se ha encontrado y se ha replicado la solicitud de retransmisión a nodos anteriores del *GPlane*.

Por ejemplo, si el campo *Confirmación* tuviera valor "01101...", indicaría que el primer paquete solicitado no se ha encontrado en el nodo que envía este mensaje *GAck*, en cambio el segundo y el tercero sí se han localizado en este nodo, el cuarto no se ha hallado, el quinto sí, y así sucesivamente. Como el tamaño del campo *Confirmación* es de 32 bits, la propuesta estará limitada a solicitudes de bloques de hasta 32 paquetes.

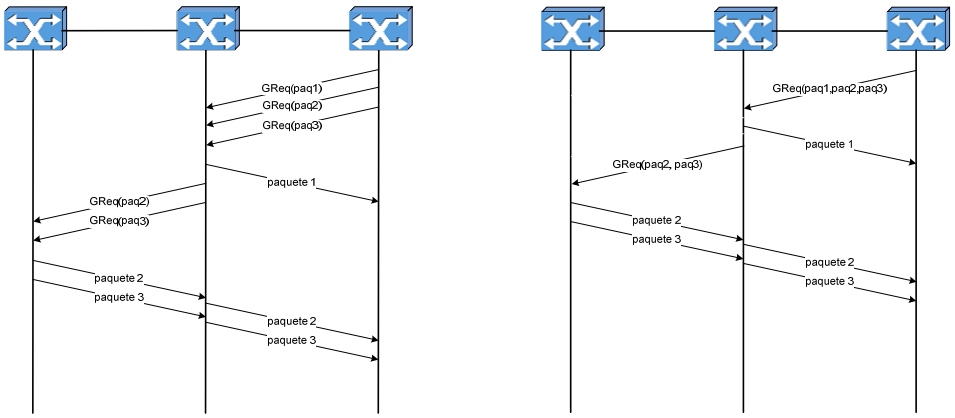

Figura 3-17. Recuperación GLRP de paquetes empleando retransmisiones simples y por bloques, respectivamente

3.8 Reordenación de paquetes

Tradicionalmente, cuando se producen pérdidas, los paquetes retransmitidos extremo a extremo se mezclan con los nuevos paquetes entrantes, provocando que el flujo MPLS se desordene. La reordenación suele ser un proceso transparente para MPLS, ya que está en manos del protocolo de transporte [26]. No obstante, se puede evitar el desorden si se emplea GLRP para recuperar paquetes, facilitando el trabajo del protocolo de transporte. En este sentido, cuando se detecta una pérdida se solicita su retransmisión local al nodo previo del *GPlane* y se empiezan a almacenar todos los paquetes nuevos que lleguen de ese flujo. Cuando el nodo reciba el paquete solicitado, lo reenvía hacia el siguiente salto, seguido de todos los paquetes almacenados previamente, para evitar que adelanten al paquete recuperado localmente. Es un mecanismo inspirado en otros esquemas de reordenación de paquetes en dominios MPLS que ya han demostrado su eficacia, como en [27].

En realidad, la reordenación de los paquetes recuperados localmente se podría analizar desde dos puntos de vista:

- Ordenación en el nodo que inicia la solicitud GLRP: Cuando se detecta el descarte de un paquete, se inicia el proceso de recuperación local y todos los nuevos paquetes se almacenarán hasta que el paquete perdido se recupere o hasta que se detecte que no se puede recobrar localmente. A partir de este momento se reenvía hacia el siguiente salto el paquete recuperado localmente (si se ha conseguido), seguido de los paquetes que se almacenaron temporalmente. Esta opción tiene la ventaja de que las recuperaciones locales son transparentes para los nodos posteriores al que advirtió la pérdida, así como para el protocolo de transporte, ya que no detectarán la pérdida ni recibirán paquetes desordenados. En cierto modo, tiene el inconveniente de que se genera un cuello de botella en el nodo de detección del descarte. Los paquetes posteriores al descartado se almacenarán temporalmente hasta que se recupere el perdido o hasta que se decida que no se va a poder recuperar localmente. Esto da lugar a un mayor retardo para los paquetes posteriores al perdido, aumentando el *jitter* de la transmisión.
- Ordenación en el *head-end*: Cuando se detecta el descarte de un paquete en un nodo intermedio, igualmente se inicia el proceso de recuperación local. Los nuevos paquetes ya no se almacenarán temporalmente en ese nodo, sino que se reenvían normalmente. Será en el nodo de salida del dominio MPLS donde se guarden temporalmente hasta conseguir que se ordenen con la llegada de los paquetes recuperados localmente. Esta estrategia tiene la ventaja de que no se originan cuellos de botella aleatorios en nodos intermedios, por lo que el *jitter* no se ve afectado. Como inconveniente, se requiere que el *head-end* sea un nodo con capacidad GLRP. Además, el nodo de salida tendrá mayor dificultad para detectar qué paquetes no se han podido re-transmitir localmente, ya que no recibe los mensajes $GAck$. Esto se podría solucionar mediante el empleo de *timeouts* para decidir qué paquetes no ha podido recuperar GLRP.

En particular, el tiempo que emplea el primer paquete fuera de orden para atravesar el LSP cuando se hace reordenación de paquetes se puede expresar como:

$$\underbrace{\sum_{i=1}^{DD-1}\delta_{i,i+1}}_{\text{A}} + \underbrace{\sum_{i=DD}^{DD-d+1}\delta_{i,i-1}}_{\text{B}} + \underbrace{\sum_{i=DD-d}^{DD-1}\delta_{i,i+1}}_{\text{C}} + \underbrace{\sum_{i=DD}^{n-1}\delta_{i,i+1}}_{\text{D}}$$

Donde:

- $\delta_{i,i+1}$: Retardo de un paquete al reenviarse desde un nodo i hasta el siguiente nodo del LSP.
- A: Retardo del paquete desde el LER de entrada hasta el nodo de detección del descarte (DD).
- B: Retardo del mensaje de solicitud de retransmisión local que se envía desde DD hasta el nodo del *GPlane*, situado a un diámetro de d saltos, que satisface la recuperación local.
- C: Retardo del paquete recuperado localmente, enviado desde el nodo del *GPlane* que satisface la recuperación local, hasta el nodo de detección del descarte (DD) que solicitó la retransmisión local.
- D: Retardo del paquete desde DD hasta el nodo LER de salida del dominio (nodo n).

3.9 GLRP en rutas punto-multipunto

Las rutas punto-multipunto responden a las necesidades de los servicios que requieren distribución de información hacia múltiples destinos desde un nodo origen, pero teniendo en cuenta ciertos aspectos:

- Hacer un uso eficiente de los recursos, evitando duplicar el tráfico.
- Proteger los flujos de datos frente a fallos.
- Administrar las rutas de distribución de los datos.
- Gestión flexible del ancho de banda, de los puntos de distribución y de las rutas.

IP Multicast es la solución clásica para el tráfico multidifusión, ya que permite que el tráfico se transporte de forma eficiente a través del árbol

multipunto simplificando, incluso, el aumento de la capacidad de la red o la adición de nuevos nodos. Hasta cierto punto, es una solución que requiere IP extremo a extremo, teniendo en cuenta que el IGP es el que establece los caminos y se necesita un protocolo para el control multipunto en el núcleo de la red. Además, en caso de fallo se debe reconstruir el árbol de distribución, por lo que el principal inconveniente de *IP Multicast* es la falta de herramientas de ingeniería de tráfico y de restauración rápida ante fallos de nodo o enlace [28].

Otra solución es la configuración de LSP punto-multipunto en dominios MPLS. Como ya se ha comentado anteriormente, las técnicas existentes de ingeniería de tráfico en MPLS aportan garantías de QoS, optimización de recursos y recuperación rápida ante fallos, pero están limitadas a túneles LSP punto a punto. Las mejoras del protocolo MPLS-TE para el soporte de LSP punto-multipunto permiten crear árboles de distribución de flujos basados en túneles con un origen y múltiples destinos, empleando para ello túneles LSP con capacidad de ingeniería de tráfico.

En este caso, se puede seguir haciendo uso de las técnicas de ingeniería de tráfico y de restauración rápida de rutas propios de MPLS. Así se alcanzan una serie de ventajas:

- No se requieren nuevos protocolos *multicast*.
- Se hace un uso eficiente de los recursos disponibles en la red.
- Elevada fiabilidad del servicio al emplear *Fast-Reroute* de RSVP-TE, ya que un mismo túnel de respaldo puede proteger, tanto a LSP punto a punto, como a LSP punto-multipunto.
- Facilidad para controlar las rutas que sigue el tráfico.
- Se mantienen los criterios de QoS característicos de MPLS.
- Flexibilidad y coste incremental durante el despliegue de la red.

Diversos estudios consideran los LSP punto-multipunto de MPLS como un mecanismo adecuado para los servicios actuales [29], como la difusión de video (IPTV), redes privadas virtuales IP o Ethernet, o para aplicaciones críticas (*Mission Critical Networking*), reduciendo el uso de ancho de banda y los costes y mejorando, además, el funcionamiento, la escalabilidad y la disponibilidad del servicio [30]. En particular, un LSP punto-multipunto con Ingeniería de Tráfico es un LSP unidireccional que tiene un único nodo LER de entrada y uno o varios LER de salida. Los servicios punto-multipunto para el reenvío de datos desde un origen a múltiples destinos se pueden proveer mediante cualquier

combinación de LSP punto a punto ó punto-multipunto, en función del grado de optimización requerido en la red, para lo que se podrán aplicar mejores técnicas de ingeniería de tráfico si lo comparamos con *IP Multicast*:

- Un LSP punto-multipunto se podrá configurar con restricciones de ingeniería de tráfico para el reenvío eficiente de paquetes a través de múltiples nodos en los que se bifurque el LSP; de forma que, aunque el reenvío de paquetes sea una tarea propia del Plano de Datos, el Plano de Control deberá garantizar que se señalizan LSP que permitan que dicho reenvío se lleve a cabo de forma eficiente. Además el *routing determinístico* permite especificar de forma explícita algunos o todos los nodos por los que debe crearse la ruta multipunto. Esto permite al operador controlar el conjunto de nodos que atraviesa cada LSP, diferenciar servicios o balancear el tráfico. Actualmente *IP Multicast* no soporta este tipo de encaminamiento de forma nativa.
- Los mecanismos de señalización de LSP punto-multipunto deben tener la posibilidad de añadir o eliminar nodos destino, soportando, además, los procesos de re-optimización de rutas propio de RSVP-TE.
- El protocolo de señalización RSVP-TE debe permitir la especificación de reservas de recursos para los LSP punto-multipunto, lo cual no está soportado por *IP Multicast*.
- Se puede proteger ante fallos el túnel punto-multipunto, empleando los mecanismos de recuperación de MPLS-TE, con el objetivo de minimizar la desconexión del servicio de difusión. En cambio, *IP Multicast* dependerá de la convergencia del IGP cuando se produzca un fallo, con períodos de recuperación que van desde cientos de milisegundos hasta varios segundos, mientras que los LSP punto-multipunto suelen tener períodos de recuperación inferiores a 50 ms.
- La reducción de la complejidad del procesamiento que llevan a cabo los nodos internos del dominio MPLS mejora la escalabilidad de todo el sistema, reduciendo además la posibilidad de fallos. Esto se consigue al disminuir el número de protocolos (y por tanto la carga de CPU y uso de memoria) que se ejecutan en los nodos intermedios. Esta simplificación se puede lograr eliminando protocolos multicast ya existentes en el núcleo, como PIM (*Protocol Independent Multicast*), ya que RSVP-TE podrá señalizar los LSP punto-multipunto.

Si bien, mientras que en las comunicaciones punto a punto el protocolo de transporte puede emplear confirmaciones positivas (Ack), transmitidas por el receptor para conocer qué paquetes han llegado correctamente a su destino, en las comunicaciones multipunto no resulta posible aplicar un esquema de ACKs generalizado. Esto implicaría que al recibir una confirmación por cada posible receptor, el protocolo resultante no sería escalable. La mayoría de los protocolos de transporte multipunto utilizan un mecanismo de confirmaciones negativas (Nack), para controlar la correcta recepción de los paquetes [31], [32]. Según este esquema, cada vez que un receptor detecte la ausencia de un paquete, debe enviar una confirmación negativa al emisor para solicitar la retransmisión. La estrategia de confirmaciones negativas, aunque aumenta la escalabilidad de los protocolos multipunto, también presenta otros problemas. Por ejemplo, cuando un paquete se descarta en un nodo intermedio y, por tanto, no llega a los receptores, se recibirá un elevado número de Nack. Esto es lo que se conoce como *implosión del emisor* o *tormenta de Nack*. Para reducir el número de confirmaciones negativas se han desarrollado diversos mecanismos:

- El uso de temporizadores en los receptores disminuye el número de Nack que el emisor recibe. Cuando un receptor detecta la falta de un paquete de información, genera un tiempo de espera al azar. Si ese tiempo transcurre sin que ningún otro receptor haya enviado un Nack, el receptor lo transmite al emisor.
- Jerarquización de receptores: Se aprovecha el esquema arbóreo de distribución de los datos para imponer una jerarquía entre todos los receptores, en la cual cada receptor sólo puede enviar un Nack a su eslabón superior.
- Estrategias colaborativas entre nodos intermedios, basadas en que los nodos realicen un proceso de filtrado o poda de Nack, de manera que, de varias solicitudes de retransmisión provenientes de varios receptores, sólo una llegue al emisor.
- Propuestas *Reliable Multicast*, en el que se habilitan ciertos nodos intermedios que actúan como repetidores del tráfico, almacenando temporalmente paquetes. Cada repetidor sirve a un conjunto de receptores, que envían sus peticiones de retransmisión a dicho receptor. Sólo si el repetidor es incapaz de satisfacer la petición de retransmisión se enviaría un Nack al emisor.

En el caso de GLRP, son los nodos intermedios que detectan la pérdida los que solicitan su retransmisión, ahorrando un tiempo destacable si se compara,

por ejemplo, con las técnicas basadas en *Reliable Multicast*, ya que en estos casos las pérdidas se siguen detectando en el receptor, con el consiguiente margen de tiempo que esto también implica por el uso del temporizador de recepción.

De ahí que, partiendo de una topología con K destinos, el numero de retransmisiones necesarias ante una pérdida en un LSP punto-multipunto es K. En cambio, al emplear GLRP, el numero de retransmisiones que se harán desde el emisor es $K*P_{emisor}$ retransmisiones, donde P_{emisor} es la probabilidad de que el paquete perdido no se pueda recuperar localmente desde algún nodo del *GPlane*. En este caso quedará en manos del protocolo de transporte la retransmisión extremo a extremo desde el emisor. Así, la mejora que aporta GLRP a la implosión del emisor ante la pérdida de un paquete es la diferencia que existe entre el número de retransmisiones desde el origen que llevaría a cabo el protocolo de transporte y el número de retransmisiones que se deberían hacer desde el origen debido a que GLRP no ha conseguido recuperar el paquete localmente desde algún nodo del *GPlane*:

K - P_{emisor} $*K = (1$ - $P_{emisor}) * K = P_{GLRP} \cdot K,$

Donde:

- K es el número de destinos o retransmisiones desde el emisor llevadas a cabo por el protocolo de transporte.
- P_{emisor} es la probabilidad de que GLRP no consiga recuperar localmente un paquete.
- P_{GLRP} es la probabilidad de que GLRP consiga recuperar un paquete.

La probabilidad de que un paquete perdido se pueda recuperar localmente desde un nodo GLRP del *GPlane* o de que se tenga que retransmitir extremo a extremo desde el emisor se analizará en detalle en el Capítulo 4.

3.10 Referencias del capítulo

[1] D. Zhang and D. Ionescu, "A new practical packet loss estimator for MPLS VPN services," The Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceedings, vol 2. Pág. 587-591. 2004. [Online]: http://dx.doi.org/10.1109/PCCC.2009.5403827 (último acceso abril 2014).

[2] C. Awad, B. Sanso and A. Girard, "Differentiated reliability in traffic engineered MPLS and DiffServ-aware next generation networks," 7th International Workshop on Design of Reliable Communication Networks, Pág. 265-272. 2009. [Online]: http://dx.doi.org/10.1109/DRCN.2009.5339998 (último acceso abril 2014).

[3] F.J. Rodríguez-Pérez, J.L. González-Sánchez and A. Gazo-Cervero, "RSVP-TE Extensions to Provide Guarantee of Service to MPLS," 6th IFIP Networking Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, volume 4479 of LNCS. Pág. 808-819. Springer-Verlag, 2007. [Online]: http://link.springer.com/chapter/10.1007%2F978-3-540-72606-7_69 (último acceso abril 2014).

[4] F.J. Rodríguez-Pérez, J.L. González-Sánchez, J. Carmona-Murillo and D. Cortés-Polo, "An OAM function to improve the packet loss in MPLS-TP domains for prioritized QoS-aware services," International Journal of Communication Systems, vol [pendiente], nº [pendiente]. Pag [pendiente]. 2013. [Online]: http://dx.doi.org/ 10.1002/dac.2742 (último acceso abril 2014).

[5] A. Pruteanu, V. Iyer and S. Dulman, "Gossip-Based Failure Estimator for Large-Scale Dynamic Networks," 20th International Conference on Computer Communications and Networks, ICCCN '11. Pág. 1-6. 2011. [Online]: http://dx.doi.org/10.1109/ICCCN.2011.6006082 (último acceso abril 2014).

[6] F.J. Rodríguez-Pérez, J.L. González-Sánchez and A. Gazo-Cervero, "GLRP Proposal to Improve Trust and Delay of MPLS Flows for MCN Services," International Journal of Computer Science and Information Security, vol 6, nº 1. Pag 75-82. 2009. [Online]: http://arxiv.org/abs/0911.0484 (último acceso abril 2014).

[7] H. Zhang, Y. Zhao, D. Han, J. Zhang and J. Xing, "QoS sensitive routing in DiffServ MPLS-TP networks," International Conference on Computer Application and System Modeling, ICCASM '10. Pág. 726-730. 2010. [Online]: http://dx.doi.org/10.1109/ICCASM.2010.5620187 (último acceso abril 2014).

[8] Weigang Wu, Jiannong Cao and Xiaopeng Fan, "Overhearing-Aided Data Caching in Wireless Ad Hoc Networks," 29th IEEE Int. Conf. on Distributed Computing Systems, ICDCS Workshops '09. Pág. 137-144. 2009. [Online]: http://dx.doi.org/10.1109/ICDCSW.2009.30 (último acceso abril 2014).

[9] A.C. Valera, W.K.G Seah and S.V Rao, "Improving Protocol Robustness in Ad Hoc Networks through cooperative packet caching and shortest multi*Path* routing," IEEE Trans on mobile computing, vol 4, nº 5. Pag 443-457. 2005. [Online]: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01492358 (último acceso abril 2014).

[10] G.T. Heineman, G. Pollice and S. Selkow, Capítulo 8: "Network Flow Algorithms" de libro "Algorithms in a Nutshell," Oreilly Media. Pag 226-250. 2008. ISBN 978-0-596-51624-6.

[11] E. Ko, D. An and H. Yoon, "Congestion control for sudden bandwidth changes in TCP," International Journal of Comm. Systems, vol 25, nº 12. Pag 1550-1567.2. [Online]: http://dx.doi.org/10.1002/dac.1322 (último acceso abril 2014).

[12] F.J. Rodríguez-Pérez, J.L. González-Sánchez, A. Gazo-Cervero and M. Castro-Ruiz, "RSVP-TE Extensions to offer Guarantee Of Service to Privileged Flows in MPLS Networks," Asian Conference on Comm. Systems and Networks. Pág. 77-83. 2007. [Online]: http://dl.acm.org/citation.cfm?id=1712884 (último acceso abril 2014).

[13] L. Andersson et al., "MPLS Transport Profile (MPLS-TP) Control Plane Framework," IETF Request for Comments 6373, Standards Track. 2011.

[14] D. Frost, S. Bryant and M. Bocci, "MPLS Transport Profile Data Plane Architecture," IETF Request for Comments 5960, Standards Track. 2010.

[15] D. Pao, X. Wang, Z. Lu, "Design of a near-minimal dynamic perfect hash function on embedded device," 15th International Conference on Advanced Communication Technology. Pág. 457-462. 2013.

[16] F.J. Rodríguez-Pérez and J.L. González-Sánchez, "Extending RSVP-TE to support Guarantee of Service in MPLS," Capítulo de libro "Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications". Pág. 149-154. Springer-Verlag, 2007. [Online]: http://dx.doi.org/10.1007/978-1-4020-6266-7_28 (último acceso abril 2014).

[17] D. Beller and A. Farrel, "An In-Band Data Communication Network For the MPLS Transport Profile," IETF RFC 5718, Standards Track. 2010.

[18] D. Beller and A. Farrel, "MPLS Transport Profile (MPLS-TP) Survivability Framework," IETF Request for Comments 6372, Standards Track. 2011.

[19] M. Monti, C.F. Tschudin and M.Luise, "Stability and Sensitivity Analysis of Traffic-Shaping Algorithms Inspired by Chemical Engineering," IEEE Journal on Selected Areas in Comm., vol 31, nº 6. Pag 1105-1114. 2013. [Online]: http://dx.doi.org/10.1109/JSAC.2013.130612 (último acceso abril 2014).

[20] Sun Xiaoling and Chen Haihong, "Research on IPv6 Routing Technology," Fifth Int. Conference on Computational and Information Sciences (ICCIS). Pag. 1409-1412. 2013. [Online]: http://dx.doi.org/10.1109/ICCIS.2013.372 (último acceso julio 2014).

[21] Lei Shi, A. Davy, D. Muldowney, S. Davy, E. Höfig and Xiaoming Fu, "Intrinsic monitoring within an IPv6 network: mapping node information to network *Path*s," Int. Conference on Network and Service Management. Pag. 370-373. 2010. [Online]: http://dx.doi.org/10.1109/CNSM.2010.5691232 (último acceso julio 2014).

[22] Yong Yu, Jia-Lei Wu and Wei Sun, "A Bandwidth Reservation Scheme in MPLS based Mobile IPv6 Network," Int. Conference on Apperceiving Computing and Intelligence Analysis. Pag. 260-264. 2008. [Online]: http://dx.doi.org/10.1109/ICACIA.2008.4770019 (último acceso julio 2014).

[23] R. Yunos, N.M. Noor and S.A. Ah, Siti Arpah, "Research on IPv6 Routing Technology," International Conference on Information Retrieval & Knowledge Management (CAMP). Pag. 204-208. 2010. [Online]: http://dx.doi.org/10.1109/INFRKM.2010.5466916 (último acceso julio 2014).

[24] A. Jayaraj, T. Venkatesh and C.S.R Murthy, "Loss classification in optical burst switching networks using machine learning techniques: improving the performance of TCP," IEEE Journal on Selected Areas in Comm., vol 26, nº 6. Pag. 45-54. 2008. [Online]: http://dx.doi.org/10.1109/JSACOCN.2008.033508 (último acceso julio 2014).

[25] A. Maach, A.S. Hafid and A. Belbekkouche, "Burst loss reduction schemes in optical burst switching networks," International Symposium on Performance Evaluation of Computer and Telecommunication Systems. Pag. 256-262. 2008.

[26] R.M. Narasiodeyar and A.P. Jayasumana, "Improvement in packet-reordering with limited re-sequencing buffers: An analysis," 38th Conference on Local Computer Networks (LCN). Pag. 416-424. 2013. [Online]: http://dx.doi.org/10.1109/LCN.2013.6761274 (último acceso julio 2014).

[27] L. Hundessa and J. Domingo-Pascual, "Reliable and fast rerouting mechanism for a protected label switched *Path*," IEEE Global Telecommunications Conference (GLOBECOM) Vol. 2. Pag. 1608-1612. 2002. [Online]: http://dx.doi.org/10.1109/GLOCOM.2002.1188469 (último acceso julio 2014).

[28] Ning Wang and G. Pavlou, "Traffic Engineered Multicast Content Delivery Without MPLS Overlay," IEEE Transactions on Multimedia, vol 9, nº 3. Pag 619-628. 2007. [Online]: http://dx.doi.org/10.1109/TMM.2006.888016 (último acceso abril 2014).

[29] I. Martinez-Yelmo, D. Larrabeiti, I. Soto and P. Pacyna, "Multicast traffic aggregation in MPLS-based VPN networks," IEEE Comm Magazine, vol 45, nº 10. Pag 78-85. 2007. [Online]: http://dx.doi.org/10.1109/MCOM.2007.4342827 (último acceso abril 2014).

[30] Jiang Zhang, S. Ruepp, M.S. Berger and H. Wessing, "Protection for MPLS-TP multicast services," 7th International Workshop on Design of Reliable Communication Networks. Pag. 297-304. 2009. [Online]: http://dx.doi.org/10.1109/DRCN.2009.5339994 (último acceso julio 2014).

[31] T. Eckert et al., "MPLS Multicast Encapsulations," IETF Request for Comments 5332, Standards Track. 2008.

[32] E. Rosen et al., "Multicast in MPLS/BGP IP VPNs," IETF Request for Comments 6513, Standards Track. 2012.

Capítulo 4. Análisis de la propuesta

La preocupación por el hombre y su destino siempre debe ser el interés primordial de todo esfuerzo técnico. Nunca olvides esto entre tus diagramas y ecuaciones.
Albert Einstein

El anterior capítulo se dedicó a analizar posibles capacidades adicionales de GLRP. En el presente capítulo se llevará a cabo el modelado de la mejora introducida por GLRP tanto en el retardo como en el consumo de recursos de red. Así, el objetivo es encontrar las ecuaciones de mejora de retardo y consumo de recursos de la red mediante GLRP, con respecto a los esquemas tradicionales de control de pérdidas extremo a extremo. También se desarrollará un análisis probabilístico de las recuperaciones GLRP en función de diversos parámetros.

4.1 Modelado de GLRP

En este capítulo se analiza la propuesta GLRP con el objetivo de comprobar la mejora que aporta con respecto a las actuales técnicas de recuperación de pérdidas extremo a extremo [1]. Para ello se calcula la diferencia existente entre la retransmisión extremo a extremo y las recuperaciones GLRP con respecto al retardo y al consumo de recursos [2]. Inicialmente se caracterizará una topología general de dominio MPLS-TP como un grafo [3]. Después, sobre dicho grafo se modelará el comportamiento de los actuales protocolos de transporte basados en retransmisiones extremo a extremo, teniendo en cuenta, tanto el coste de los enlaces (retardo o recursos consumidos), como un conjunto de restricciones que limitarán la factibilidad de cada caso del problema [4]. Luego se modelará el comportamiento de las recuperaciones GLRP empleando el mismo modelo, para que se puedan comparar ambos esquemas bajo las mismas premisas. Finalmente,

una vez equiparados ambos modelos, se calcularán las diferencias entre ambos, con el objetivo de encontrar la ecuación que permita extrapolar este estudio a otras topologías y muestras de tráfico.

Sea un flujo $\varphi(G)$ en el dominio MPLS-TP $G(U)$ a través de un camino $LSP_{i,n}$, con origen en el nodo x_i y destino en el nodo x_n con capacidad GLRP, para $\{i, n\} \in \mathrm{N}$ (ver Figura 4-1). En este entorno, si x_n sólo comprueba la etiqueta MPLS de entrada o bien el puerto por el que recibe los paquetes, sólo podrá reconocer a x_{n-1} como posible emisor del flujo de datos $\varphi(x_i, x_n)$. En consecuencia, si x_{n-1} introduce algún mecanismo de agregación de k flujos, se obtiene:

$$\varphi(x_{n-1}, x_n) = \sum_{i=1}^{k} \varphi_i (x_{n-1}, x_n)$$

Es decir, el flujo de datos en realidad puede estar formado por paquetes de k flujos agregados. Por este motivo, el análisis del campo *etiqueta MPLS* no se considera una estrategia fiable a la hora de descubrir los nodos por los que han pasado los paquetes de datos. Por otro lado, como se ha comentado anteriormente, en caso de congestión se pueden producir pérdidas de paquetes a lo largo de la ruta. Por ejemplo, si se produce el descarte de uno o varios paquetes en x_n, dejaría de cumplirse la Condición de Conservación del Flujo en la red, provocando que el flujo de paquetes salientes de x_n sea menor que el entrante:

$$\sum_{j=1}^{k} p_{nj} < \sum_{i=1}^{k} p_{in} \ ,$$

siendo p_{ij} el volumen de tráfico enviado desde x_i al nodo x_j.

En este caso, GLRP permite que x_n asuma el hecho de que el tráfico perdido haya podido quedar almacenado en otros nodos GLRP anteriores en $LSP_{i,n}$. El primero al que podrá enviar una solicitud de retransmisión local será el nodo GLRP anterior más cercano en el *GPlane* que, como se ha comentado anteriormente, es un subconjunto de nodos de $LSP_{i,n}$ con capacidad GLRP, estando cada uno de ellos a una distancia o *diámetro d* de x_n:

$GPlane \subseteq LSP_{i,n} = \{(x_i, x_{i+1}), (x_{i+1}, x_{i+2}), \ldots, (x_{n-1}, x_n)\} \subseteq U$, con $d(x_i, x_n) = n\text{-}i$, / x_i es GLRP, $\forall\ x_i \in LSP_{i,n}$

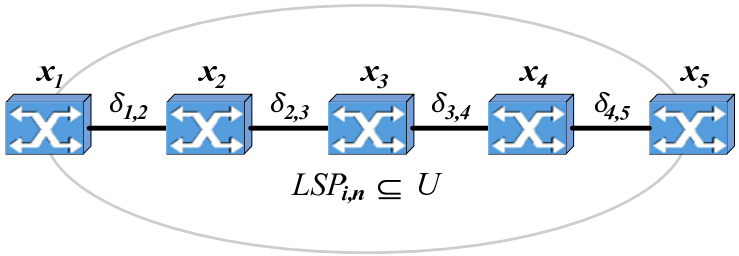

Figura 4-1. Esquema de la ruta $LSP_{i,n}$ para i=1 y n=5

Así, en caso de pérdida de paquetes, x_n puede solicitar una retransmisión a los nodos pertenecientes a *GPlane*, con el objetivo de evitar retransmisiones extremo a extremo desde x_i, lo que provocaría un incremento del $\varphi(G)$ global en el dominio $G(U)$.

4.2 Análisis del retardo

El retardo mínimo que emplea un paquete al viajar entre dos nodos cualesquiera de nuestro dominio *U(G)* se puede obtener a partir del algoritmo de mínimo coste:

$$min\ \delta(x_i,\ x_j) = \sum_{i=1}^{n}\sum_{j=1}^{n} \delta_{ij}\, x_{ij}$$

sujeto a:

$$\sum_{l=2}^{n} x_{1l} = 1$$

$$\sum_{i=1}^{n} x_{il} - \sum_{j=1}^{n} x_{lj} = 0,\ l = 2,3,...,n-1$$

$$\sum_{l=1}^{n-1} x_{ln} = 1$$

Donde: $x_{i,j} = 1, \forall (x_i, x_j) \in LSP_{i,n}$, $x_{i,j} = 0, \forall (x_i, x_j) \notin LSP_{i,n}$, $\delta_{i,i} = 0, \forall i$

A partir de aquí se pueden definir una serie de parámetros de interés:

- $\Delta_{e\text{-}e}(x_i, x_j)$ es el retardo total de obtención de un paquete en x_j, proviniendo de x_i, empleando recuperaciones extremo a extremo ante posibles pérdidas del paquete.
- $\Delta_d(x_i, x_j)$ es el retardo total de obtención de un paquete en x_j, proviniendo de x_i, empleando recuperaciones locales con diámetro d ante posibles pérdidas del paquete.
- $TDD_{e\text{-}e}(x_i, x_n)$ es el tiempo que transcurre hasta que el nodo extremo x_n detecta la pérdida de un paquete que proviene de x_i.
- $TDD_d(x_i, x_j)$ es el tiempo que transcurre hasta que el nodo intermedio x_j detecta la pérdida de un paquete que proviene de x_i.
- $\delta_{e\text{-}e}(x_i, x_j)$ es el tiempo empleado en recuperar extremo a extremo un paquete que provenía de x_i y descartado en x_j.
- $\delta_d(x_i, x_j)$ es el tiempo empleado en recuperar localmente, con diámetro d, un paquete que provenía de x_i y descartado en x_j.

En particular, si x_n no es un nodo con capacidad GLRP, para cada paquete descartado del flujo de paquetes $\varphi(x_i, x_n)$, la función *Tiempo de Detección de Descarte (TDD)* se define como:

$$TDD_{e2e}(x_i, x_n) = \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

En este caso se emplearían retransmisiones extremo a extremo, ya que x_n es un nodo sin capacidad GLRP. Así, el tiempo empleado en la retransmisión de un paquete perdido se obtiene como:

$$\delta_{e2e}(x_i, x_n) = 2 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

Por tanto, el tiempo total de obtención del paquete descartado en x_n desde el instante inicial de su transmisión es:

$$\Delta_{e-e}(x_i, x_n) = TDD_{e\text{-}e}(x_i, x_n) + \delta_{e\text{-}e}(x_i, x_n)$$

$$\Delta_{e-e}(x_i, x_n) = \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

$$\Delta_{e-e}(x_i, x_n) = 3 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

En cambio, si x_n sí es un nodo con capacidad GLRP (ver Figura 4-2) y, además, se trata del nodo destino ($DD=n$), para cada paquete descartado del flujo $\varphi(x_i, x_n)$ la función TDD se puede obtener como:

$$TDD_d(x_i, x_n) = \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

Además, el tiempo empleado en su retransmisión local con diámetro $d \in \mathrm{N}$, se define como:

$$\delta_d(x_i, x_n) = 2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}\, x_{l,l+1}\,,$$

sujeto a: $0 < d < n - i$

La restricción $d < n - i$ es necesaria, ya que si $d = n\text{-}i$, entonces:

$l = n\text{-}d = n\text{-}(n\text{-}i) = n\text{-}n + i = i$

y se obtendría que:

$$2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}\, x_{l,l+1} = 2 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

Es decir, se trataría de una retransmisión extremo a extremo. Por otro lado, si se considera $d>n\text{-}i$ se estaría intentando conseguir una retransmisión desde un nodo anterior a x_i, siendo éste el emisor del flujo $\varphi(x_i, x_n)$, lo cual no es factible.

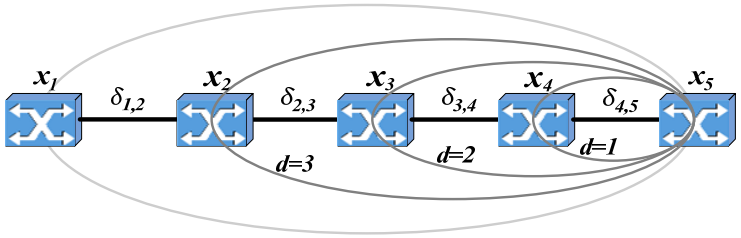

Figura 4-2. Recuperaciones locales factibles para $DD = x_n = 5$

Así, el tiempo total empleado para obtener en el destino x_n, el paquete descartado desde el instante inicial de su transmisión es:

$\Delta_d(x_i, x_n) = TDD_d(x_i, x_n) + \delta_d(x_i, x_n)$:

$$\Delta_d(x_i, x_n) = \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

En este punto interesa demostrar que $\Delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$, es decir:

$$\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}\, x_{l,l+1} < 3 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

$$\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}\, x_{l,l+1} < \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

$$2 \sum_{l=n-d}^{n-1} \delta_{l,l+1}\, x_{l,l+1} < 2 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

Así pues, solo es necesario comprobar que $\delta_d(x_i, x_n) < \delta_{e\text{-}e}(x_i, x_n)$. Los miembros de esta desigualdad sólo se distinguen por una única condición: el conjunto de valores que puede tomar la variable l, por lo que habría que demostrar que l toma un menor número de valores en $\delta_d(x_i, x_n)$ que en $\delta_{e\text{-}e}(x_i, x_n)$:

$n - 1 - (n - d) < n - 1 - i$

donde:

$n - 1 - (n - d)$ es el rango de valores de l para $\delta_d(x_i, x_n)$.
$n - 1 - i$ es el rango de valores de l para $\delta_{e\text{-}e}(x_i, x_n)$.

Al hacer la comprobación se obtiene que $d < n - i$:

$n - 1 - (n - d) < n - 1 - i$
$n - 1 - n + d < n - 1 - i$
$-1 + d < n - 1 - i$
$-1 + 1 + d < n - i$
$d < n - i$

El problema se mantiene en su zona de factibilidad ya que $d < n - i$ es una de las restricciones del mismo, con lo cual queda demostrado que $\Delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$. Además, esto implica que el beneficio que obtiene

GLRP en el retardo debe ser positivo y se puede medir restando ambos miembros de la inecuación:

$$\Delta_{e-e}(x_i,x_n)-\Delta_d(x_i,x_n)=3\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}-\left(\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}+2\sum_{l=n-d}^{n-1}\delta_{l,l+1}\,x_{l,l+1}\right)=$$

$$=3\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}-\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}-2\sum_{l=n-d}^{n-1}\delta_{l,l+1}\,x_{l,l+1}=$$

$$=2\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}-2\sum_{l=n-d}^{n-1}\delta_{l,l+1}\,x_{l,l+1}=2\left(\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}-\sum_{l=n-d}^{n-1}\delta_{l,l+1}\,x_{l,l+1}\right)$$

$$\Delta_{e-e}(x_i,x_n)-\Delta_d(x_i,x_n)=2\sum_{l=i}^{n-d-1}\delta_{l,l+1}\,x_{l,l+1}$$

Así mismo, el beneficio obtenido con respecto al número de saltos es:
$2\ (n\text{-}1\text{-}i\text{-}((n\text{-}1)\text{-}(n\text{-}d)))=2\ (n\text{-}1\text{-}i\text{-}(n\text{-}1\text{-}n+d))=2\ (n\text{-}1\text{-}i\text{-}n+1+n\text{-}d)=2\ (n\text{-}d\text{-}i)$

Por otro lado, si suponemos un nodo intermedio x_{DD} ($i<DD<n$), con capacidad GLRP (ver Figura 4-3), para cada paquete descartado de $\varphi(x_i, x_n)$, la función *Tiempo de Detección de Descarte TDD* se define como:

$$TDD_d(x_i,x_{DD})=\sum_{l=i}^{DD-1}\delta_{l,l+1}\,x_{l,l+1}$$

Además, el tiempo empleado en su retransmisión local con diámetro d se define como:

$$\delta_d(x_i,x_{DD})=2\sum_{l=DD-d}^{DD-1}\delta_{l,l+1}\,x_{l,l+1}\ ,$$

sujeto a: $0 < d \leq DD - i$

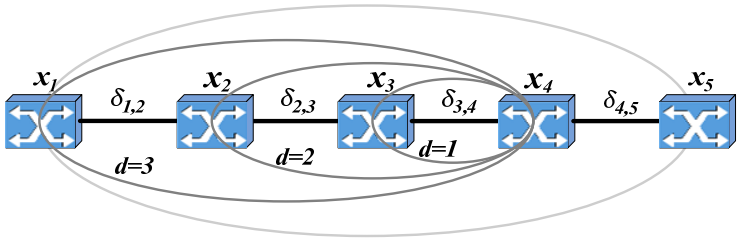

Figura 4-3. Recuperaciones locales factibles para $DD = 4 < x_n$

La restricción es necesaria, ya que si $d > DD - i$ entonces el objetivo sería conseguir una retransmisión desde un nodo anterior a x_i, siendo éste el que genera $\varphi(x_i, x_n)$, lo cual no es factible. Además, en este caso la retransmisión local desde el nodo inicial ($d=DD-i$), también aporta mejora con respecto a *e-e*, ya que x_{DD} es un nodo anterior a x_n, con lo que:

Si $DD < n \Rightarrow DD - i < n - i$.

Por consiguiente, el tiempo total para obtener en x_n un posible paquete perdido desde el instante inicial de su transmisión se calcula como:

$$\Delta_d(x_i,x_n) = TDD_d(x_i,x_{DD}) + \delta_d(x_i,x_{DD}) + \sum_{l=DD}^{n-1}\delta_{l,l+1}\,x_{l,l+1} =$$

$$= \sum_{l=i}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} + \sum_{l=DD}^{n-1}\delta_{l,l+1}\,x_{l,l+1} =$$

$$= \sum_{l=i}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} + \sum_{l=DD}^{n-1}\delta_{l,l+1}\,x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} = TDD_{e-e}(x_i,x_n) + \delta_d(x_i,x_{DD})$$

En este punto también interesa demostrar que $\Delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$:

$$TDD_{e-e}(x_i,x_n) \;+\; \delta_d(x_i,x_{DD}) < \Delta_{e-e}(x_i,x_n)$$

$$\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} \;<\; 3\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}$$

$$\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} \;<\; \sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1} + 2\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}$$

$$2\sum_{l=DD-d}^{DD-1}\delta_{l,l+1}\,x_{l,l+1} \;<\; 2\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}$$

Nuevamente sólo habría que comprobar que $\delta_d(x_i, x_{DD}) < \delta_{e\text{-}e}(x_i, x_n)$ y de nuevo la única condición que distingue a los miembros de la anterior desigualdad es el conjunto de valores que puede tomar la variable l. Por lo tanto, sólo es necesario demostrar que l toma un menor número de valores en $\delta_d(x_i, x_{DD})$ que en $\delta_{e\text{-}e}(x_i, x_n)$:

$DD - 1 - (DD - d) < n - 1 - i$

donde:

$DD - 1 - (DD - d)$ es el rango de valores de l para $\delta_d(x_i, x_{DD})$.

$n - 1 - i$ es el rango de valores de l para $\delta_{e\text{-}e}(x_i, x_n)$.

Por eso, al hacer la comprobación se obtiene de nuevo que $d < n - i$:

$DD - 1 - (DD - d) < n - 1 - i$
$DD - 1 - DD + d < n - 1 - i$
$-1 + d < n - 1 - i$
$-1 + 1 + d < n - i$
$d < n - i$,

Partiendo de la restricción $d \leq DD - i$ y conociendo además que en este caso $DD < n$, entonces también se cumple que $d < n - i$:

$$\left.\begin{array}{l} d \leq DD - i \\ \wedge \\ DD < n \end{array}\right\} \Rightarrow d < n - i$$

De nuevo el problema se mantiene en su zona de factibilidad, con lo cual queda demostrado que $\Delta_d(x_i, x_n) < \Delta_{e-e}(x_i, x_n)$, siendo $i<DD<n$. Esto también implica que el beneficio que obtiene GLRP en el retardo total debe ser positivo y se puede medir restando ambos miembros de la inecuación:

$$\Delta_{e-e}(x_i, x_n) - \Delta_d(x_i, x_n) =$$

$$= 3\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} - \left(\sum_{l=i}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} + \sum_{l=DD}^{n-1} \delta_{l,l+1}\, x_{l,l+1} \right) =$$

$$= 3\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} - \left(\sum_{l=i}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} + \sum_{l=DD}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} \right) =$$

$$= \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} - \left(\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2\sum_{l=DD-d}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} \right) =$$

$$= \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} + 2\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} - \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} - 2\sum_{l=DD-d}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} =$$

$$= 2\left(\sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} - \sum_{l=DD-d}^{DD-1} \delta_{l,l+1}\, x_{l,l+1} \right)$$

$$\Delta_{e2e}(x_i, x_n) - \Delta_d(x_i, x_n) = 2\left(\sum_{l=i}^{DD-d-1} \delta_{l,l+1}\, x_{l,l+1} + \sum_{l=DD}^{n-1} \delta_{l,l+1}\, x_{l,l+1} \right)$$

Así mismo, el beneficio obtenido con respecto al número de saltos necesarios es similar al caso anterior, ya que:

$2\ (\ n - 1 - \mathrm{i} - (\ (DD - 1) - (DD - d)\ )\ ) =$

$2\ (\ n-1-i-(DD-1-DD+d)\ ) =$
$2\ (n-1-i-DD+1+DD-d) =$
$2\ (n-d-i)$

Por tanto, el número de iteraciones de GLRP es independiente de la función *TDD* o del nodo en que se detecte el descarte. Sólo dependerá de manera directa del *diámetro* de la recuperación local que satisfaga la retransmisión local.

Por lo que sigue, también interesa analizar el problema si existen pérdidas en múltiples nodos intermedios x_{DD}. (ver Figura 4-4). En este caso, partiendo del retardo total $\Delta_d(x_i, x_n)$ ya conocido, se puede analizar el tiempo total necesario para obtener en x_n un paquete que se descarta en los n-1 últimos nodos de $LSP_{i,n}$:

$$\Delta_d(x_i,x_n) = \Delta_d(x_i,x_{n-1}) + \left[TDD_{e-e}(x_i,x_n) - TDD_d(x_i,x_{n-1})\right] + \delta_d(x_i,x_n) =$$

$$= \Delta_d(x_i,x_{n-2}) + \left[TDD_d(x_i,x_{n-1}) - TDD_d(x_i,x_{n-2})\right] + \delta_d(x_i,x_{n-1}) + \left[TDD_{e-e}(x_i,x_n) -\right.$$

$$\left. - TDD_d(x_i,x_{n-1})\right] + \delta_d(x_i,x_n) = \Delta_d(x_i,x_{n-3}) + \left[TDD_d(x_i,x_{n-2}) - TDD_d(x_i,x_{n-3})\right] +$$

$$+ \delta_d(x_i,x_{n-2}) + \left[TDD_d(x_i,x_{n-1}) + TDD_d(x_i,x_{n-2})\right] + \delta_d(x_i,x_{n-1}) + \left[TDD_{e-e}(x_i,x_n) -\right.$$

$$\left. - TDD_d(x_i,x_{n-1})\right] + \delta_d(x_i,x_n) = \ldots = \Delta_d(x_i,x_{i+1}) + \left[TDD_d(x_i,x_{i+2}) - TDD_d(x_i,x_{i+1})\right] +$$

$$+ \delta_d(x_i,x_{i+2}) + \left[TDD_d(x_i,x_{i+3}) - TDD_d(x_i,x_{i+2})\right] + \delta_d(x_i,x_{i+3}) + \left[TDD_d(x_i,x_{i+4}) -\right.$$

$$\left. - TDD_{e-e}(x_i,x_{i+3})\right] + \delta_d(x_i,x_{i+4}) + \ldots + \left[TDD_{e-e}(x_i,x_n) - TDD_d(x_i,x_{n-1})\right] + \delta_d(x_i,x_n) =$$

$$= \Delta_d(x_i,x_i) + \left[TDD_d(x_i,x_{i+1}) - TDD_d(x_i,x_i)\right] + \delta_d(x_i,x_{i+1}) + \left[TDD_d(x_i,x_{i+2}) -\right.$$

$$\left. - TDD_d(x_i,x_{i+1})\right] + \delta_d(x_i,x_{i+2}) + \left[TDD_d(x_i,x_{i+3}) - TDD_d(x_i,x_{i+2})\right] + \delta_d(x_i,x_{i+3}) + \ldots +$$

$$+ \left[TDD_{e-e}(x_i,x_n) - TDD_d(x_i,x_{n-1})\right] + \delta_d(x_i,x_n) =$$

$$= \delta_d(x_i,x_{i+1}) + \delta_d(x_i,x_{i+2}) + \delta_d(x_i,x_{i+3}) + \ldots + \delta_d(x_i,x_n) + TDD_{e-e}(x_i,x_n)$$

$$\Delta_d(x_i,x_n) = \sum_{k=i+1}^{n} \delta_d(x_i,x_k) + TDD_{e-e}(x_i,x_n)$$

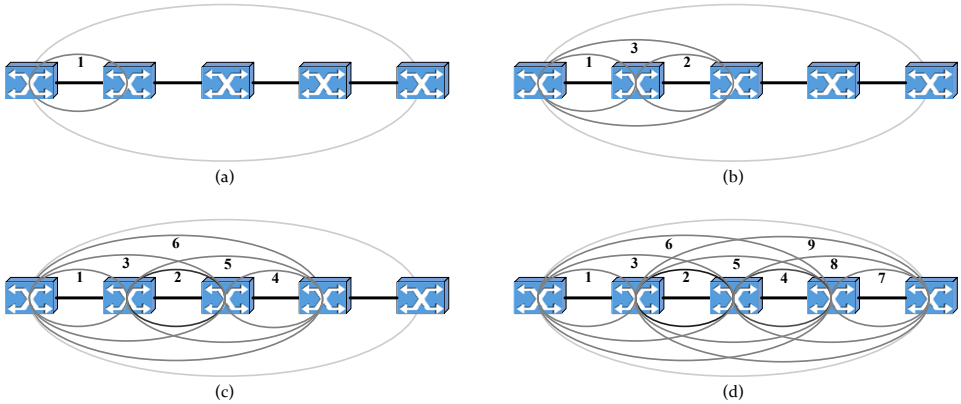

Figura 4-4. Saltos factibles en la recuperación local de un paquete perdido en varios nodos de $LSP_{i,n}$: (a) $x_{DD} = x_2$; (b) $x_{DD} = x_3$; (c) $x_{DD} = x_4$; (d) $x_{DD} = x_5$

Así, una vez definida la función $\Delta_d(x_i, x_n)$, se comprobará que:

$\Delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$, cuando se producen pérdidas en los n-i últimos nodos:

$$TDD_{e-e}(x_i, x_n) + \sum_{k=i+1}^{n} \delta_d(x_i, x_k) < 3\,(n-i) \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}$$

$$\underbrace{TDD_{e-e}(x_i, x_n)}_{A} + \sum_{k=i+1}^{n} \underbrace{\left(2 \sum_{l=k-d}^{k-1} \delta_{l,l+1}\, x_{l,l+1} \right)}_{B} < \sum_{k=i+1}^{n} \underbrace{\left(3 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1} \right)}_{C},$$

En resumen, se comprobará que A + (n-i) B < (n-i) C. Además, ya se ha demostrado que $\Delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$, por esta razón:

$TDD_{e\text{-}e}(x_i, x_n) + \delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$

$$\underbrace{TDD_{e-e}(x_i, x_n)}_{A} + \underbrace{2 \sum_{l=k-d}^{k-1} \delta_d(x_i, x_k)}_{B} < \underbrace{3 \sum_{l=i}^{n-1} \delta_{l,l+1}\, x_{l,l+1}}_{C}, \text{ para } DD{=}k$$

$A + B < C, \ \forall\, k \in \{i{+}1, i{+}2, \ \ldots, \ n\}$

Es decir, el problema se puede plantear mediante el siguiente sistema de ecuaciones:

$$\left.\begin{array}{l} A + (n\text{-}i)\ B < (n\text{-}i)\ C \\ A + B < C,\ \forall\, k \in \{i+1, i+2,\ \ldots,\ n\} \end{array}\right\}$$

$$A + (n\text{-}i)*B < (n\text{-}i)*(A+B)$$
$$A + (n\text{-}i)*B < (n\text{-}i)*A + (n\text{-}i)*B$$
$$A < (n\text{-}i)*A,\ \forall\, A \,/\, (n\text{-}i) > 1$$

Según las restricciones del problema:

$0 < d \leq DD\text{-}i < n\text{-}i$, para $\{d, DD, i, n\} \in \boldsymbol{N}$, lo que implica que $n\text{-}i>1$, que mantiene el problema en su zona de factibilidad. Por todo ello se ha comprobado que $\Delta_d(x_i, x_n) < \Delta_{e\text{-}e}(x_i, x_n)$, lo que también implica que el beneficio que obtiene GLRP en el retardo total debe ser positivo y se puede medir restando ambos miembros de la inecuación:

$$\sum_{k=i+1}^{n}\left(3\sum_{l=i}^{n-1}\delta_{l,l+1}\,x_{l,l+1}\right)-\left(\sum_{k=i+1}^{n}\left(2\sum_{l=k-d}^{k-1}\delta_{l,l+1}\,x_{l,l+1}\right)+TDD_{e-e}(x_i,x_n)\right)$$

Así mismo, el beneficio obtenido con respecto al número de saltos necesarios es similar a los casos anteriores, con la salvedad de que se repetirá $n–i$ veces: $(n - i) * 2 * (n - d - i)$. En cualquier caso, el total de posibilidades que tendrá un nodo GLRP para recuperar localmente un paquete perdido en los $n\text{-}i$ últimos nodos, vendrá dado por:

$$Max\ iteraciones = \left(\frac{(n-i)+1}{2}(n-i)\right)-1,$$

Dada la progresión aritmética limitada {1, 2, 3, ..., $n\text{-}i$}, la suma de sus términos es igual a la semisuma de los extremos multiplicada por el número de términos; a lo cual se le resta 1 ya que desde n el valor de d sólo es factible hasta i+1. Para el ejemplo de la

, con n=5, i=1 y descarte en los 4 últimos nodos, se define la progresión {1, 2, 3, 4}, para la que el número máximo de iteraciones posibles es:

$$Max\ iteraciones = \left(\frac{(5-1)+1}{2}(5-1)\right)-1 = 9$$

4.3 Análisis del consumo de recursos

Hasta este punto se ha hecho un análisis sobre el tiempo necesario para que los paquetes perdidos alcancen el nodo destino, cuando se descartan una o varias veces en diferentes nodos del túnel LSP. Por otra parte, también es interesante una comparativa acerca de la carga o consumo de recursos que implica sobre la red la retransmisión de estos paquetes perdidos [5], [6]. Para ello, teniendo en cuenta el mismo dominio de red *G(U)* que anteriormente, la ruta con mayor ancho de banda disponible entre dos nodos cualesquiera del dominio, se puede obtener como:

$$Max\ (\ \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{\beta_{ij}}x_{ij}\ )$$

sujeto a:

$$\sum_{l=2}^{n} x_{1l} = 1$$

$$\sum_{i=1}^{n} x_{il} - \sum_{j=1}^{n} x_{lj} = 0,\ l = 2,3,...,n-1$$

$$\sum_{l=1}^{n-1} x_{ln} = 1$$

donde:

$\beta_{i,j}$ es la reserva de recursos para el enlace (x_i, x_j).

$\frac{1}{\beta_{ij}}$ es el coste que representa los recursos disponibles en el enlace (x_i, x_j).

$$x_{i,j} = 1,\ \forall (x_i, x_j) \in LSP_{i,n};\qquad x_{i,j} = 0,\ \forall (x_i, x_j) \notin LSP_{i,n};\qquad \frac{1}{\beta_{i,i}} = \infty,\ \forall i$$

A partir de aquí se pueden definir una serie de parámetros de interés:

- $ABW_{e\text{-}e}(x_i, x_j)$ es el ancho de banda total acumulado al obtener un paquete en x_j, proviniendo de x_i y empleando recuperaciones extremo a extremo ante una pérdida.
- $ABW_d(x_i, x_j)$ es la carga acumulada para obtener un paquete en x_j, proviniendo de x_i, empleando recuperaciones locales con diámetro d ante una pérdida.
- $ABW_{fw}(x_i, x_n)$ es el tiempo transcurrido hasta la detección de un descarte en el nodo extremo x_n de un paquete que proviene de x_i.

– $ABW_{ret}(x_i, x_j)$ son los recursos consumidos al recuperar un paquete que provenía de x_i y descartado en x_j.

En particular, si x_n no es un nodo con capacidad GLRP, para cada paquete descartado del flujo de paquetes $\varphi(x_i, x_n)$, los recursos consumidos para su reenvío desde x_i hasta x_n son los siguientes:

$$ABW_{fw}(x_i, x_n) = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

Así mismo, los recursos consumidos al retransmitir extremo a extremo el paquete perdido se puede obtener como:

$$ABW_{ret}(x_i, x_n) = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1} = ABW_{fw}(x_i, x_n)$$

Por tanto, los recursos consumidos por cada paquete descartado en x_n no siendo éste un nodo GLRP es $ABW_{fw} + ABW_{ret}$ (ver Figura 4-5):

$$ABW_{e2e}(x_i, x_n) = 2 \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

En cambio, si x_n sí es un nodo con capacidad GLRP y, además, se trata del nodo destino ($DD=n$), para cada paquete descartado del flujo $\varphi(x_i, x_n)$, los recursos consumidos para el envío desde x_i hasta x_n son los siguientes:

$$ABW_{fw}(x_i, x_n) = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

Tras la decisión de retransmisión local con diámetro d, los recursos que consume el paquete retransmitido localmente es:

$$ABW_{ret}(x_i, x_n) = \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

sujeto a: $0 < d < n - i$

La restricción es necesaria, ya que si $d = n\text{-}i$, entonces si $l=n\text{-}d = n\text{-}(n\text{-}i) = n\text{-}n+ i = i$ y se tendría que $\sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$, es decir, se trataría de una retransmisión extremo a extremo. Por otro lado, si $d > n\text{-}i$ entonces se intentaría conseguir una retransmisión desde un nodo anterior a x_i, siendo éste el que genera $\varphi(x_i, x_n)$, lo cual no es factible.

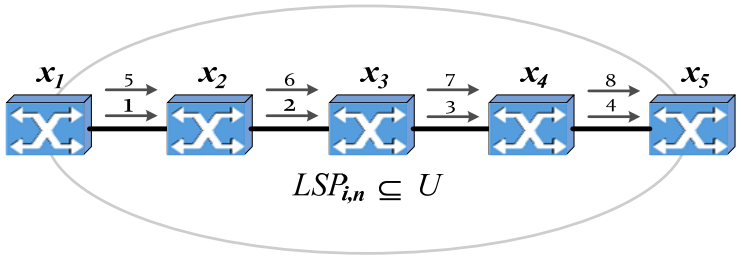

Figura 4-5. Ruta seguida por un paquete descartado en x_n y retransmitido extremo a extremo

Entonces, los recursos consumidos por cada paquete descartado en x_n siendo éste un nodo GLRP (ver Figura 4-6), es $ABW_{fw} + ABW_{ret}$:

$$ABW_d(x_i, x_n) = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

En este punto interesa demostrar que $ABW_d(x_i, x_n) < ABW_{e\text{-}e}(x_i, x_n)$:

$$\sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} < 2 \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

$$\sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} < \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

$$\sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} < \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

La única condición que distingue ambas partes de la desigualdad es el conjunto de valores que puede tomar la variable l: $n–1–(n–d) < n–1–i$. Por tanto, se comprobará que l toma un menor número de valores en el caso de retransmisión con diámetro d que en el caso de retransmisión extremo a extremo:

$n - 1 - (n - d) < n - 1 - i$
$n - 1 - n + d < n - 1 - i$
$-1 + d < n - 1 - i$
$-1 + 1 + d < n - i$
$d < n - i,$

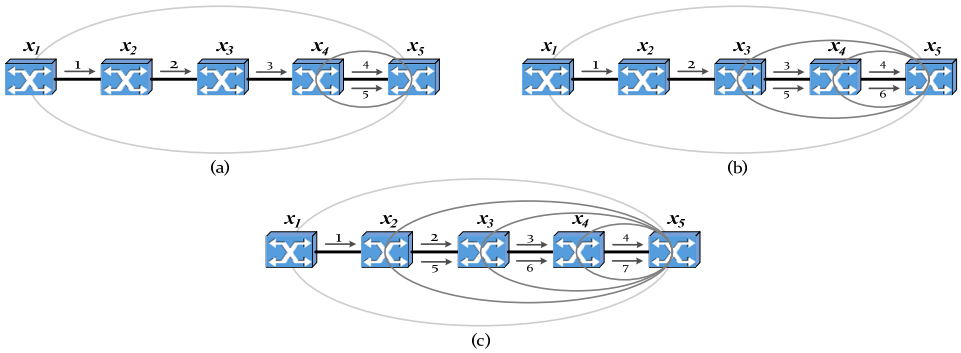

Figura 4-6. Recuperaciones locales desde el nodo extremo con capacidad GLRP x_n, con $DD = x_n = 5$:
(a) para d=1; (b) para d=2; (c) para d=3

El problema se mantiene en su zona de factibilidad ya que $d < n - i$ es una de las restricciones del mismo, con lo que $ABW_d(x_i, x_n) < ABW_{e\text{-}e}(x_i, x_n)$. Por añadidura, esto implica que el beneficio que obtiene GLRP en los recursos consumidos debe ser positivo y se puede obtener restando ambos miembros de la inecuación:

$$ABW_{e2e}(x_i, x_n) - ABW_d(x_i, x_n) = 2\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \left(\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}\right) =$$

$$= 2\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1} =$$

$$= \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1} =$$

$$= \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}$$

$$ABW_{e2e}(x_i, x_n) - ABW_d(x_i, x_n) = \sum_{l=i}^{n-d-1}\beta_{l,l+1}\,x_{l,l+1}$$

En este caso el beneficio obtenido de $ABW_{e\text{-}e}(x_i, x_n) - ABW_d(x_i, x_n)$ se puede calcular como:
$n\text{-}1\text{-}i\text{-}(n\text{-}1\text{-}(n\text{-}d)) = n\text{-}1\text{-}i\text{-}(n\text{-}1\text{-}n+d) = n\text{-}1\text{-}i\text{-}n+1+n\text{-}d = n\text{-}d\text{-}i$

Por otro lado, si suponemos un nodo intermedio x_{DD} ($i<DD<n$), con capacidad GLRP (ver Figura 4-7), para cada paquete descartado del flujo $\varphi(x_i,x_n)$, los recursos consumidos en el reenvío desde x_i hasta su pérdida en x_{DD} es el siguiente:

$$ABW_{fw}(x_i, x_{DD}) = \sum_{l=i}^{DD-1} \beta_{l,l+1}\, x_{l,l+1}$$

En el caso de que el nodo intermedio Si x_{DD} no tenga capacidad GLRP, tras la decisión de retransmisión extremo a extremo, los recursos consumidos por el paquete retransmitido desde x_i serán:

$$ABW_{ret}(x_i, x_n) = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

Entonces, resulta que el ABW total para cada paquete perdido en x_{DD} no siendo éste un nodo GLRP es:

$$ABW_{e2e}(x_i, x_n) = \sum_{l=i}^{DD-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

En cambio, si x_{DD} sí tiene capacidad GLRP, tras la decisión de retransmisión local los recursos consumidos por un paquete retransmitido localmente desde x_{DD-d} son:

$$BW_{ret}(x_i, x_n) = \sum_{l=DD-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

sujeto a: $0 < d \leq DD - i$

Como en casos anteriores, la restricción en esta situación también es necesaria, ya que si $d>DD-i$ entonces se intentaría conseguir una retransmisión desde un nodo anterior a x_i, siendo éste el que genera $\varphi(x_i, x_n)$, lo cual no es factible. Además, en este caso la decisión de retransmisión al nodo inicial ($d = DD - i$) también aporta mejora respecto a *e-e*, porque x_{DD} es un nodo anterior a x_n, es decir:

$DD - i < n - i$

De manera que, los recursos consumidos para cada paquete descartado en el nodo intermedio x_{DD}, siendo éste un nodo GLRP se puede obtener como:

$$ABW_{d}(x_i, x_n) = \sum_{l=i}^{DD-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=DD-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

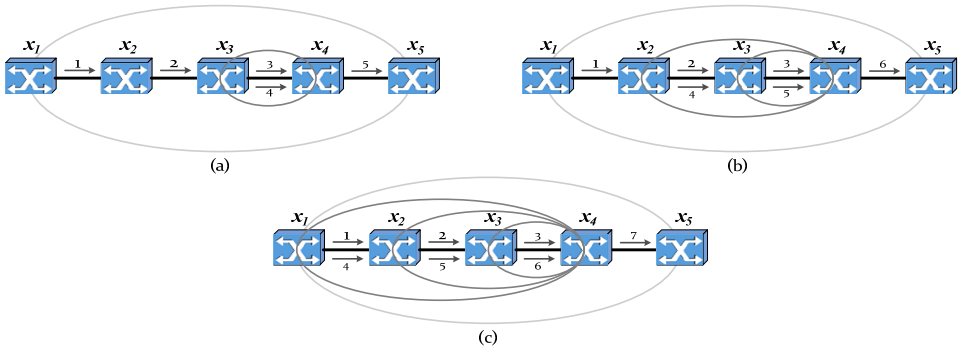

Figura 4-7. Recuperaciones locales factibles desde el nodo intermedio con capacidad GLRP $x_{DD} = x_4 < x_n$:
(a) para d=1; (b) para d=2; (c) para d=3

En este punto interesa comprobar que $ABW_d\ (x_i,\ x_n) < ABW_{e\text{-}e}\ (x_i,\ x_n)$:

$$\sum_{l=i}^{DD-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=DD-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} < \sum_{l=i}^{DD-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

$$\sum_{l=DD-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} < \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

De nuevo la única condición que distingue ambas partes de la desigualdad es el conjunto de valores que puede tomar la variable l: n-1-(DD-d) < n-1-i. Por tanto, sólo es necesario demostrar que l toma un menor número de valores en el caso de recuperación local con diámetro d que en el caso extremo a extremo:

$n - 1 - DD + d < n - 1 - i$
$n - 1 - DD + d - n + 1 < - i$
$-DD + d < - i$
$DD - d > i$

Al hacer la comprobación el problema se mantiene en su zona de factibilidad, con lo cual queda demostrado que $ABW_d\ (x_i,\ x_n) < ABW_{e\text{-}e}\ (x_i,\ x_n)$, siendo $d<DD\text{-}i$. Esto también implica que el beneficio que obtiene GLRP en el consumo de recursos total debe ser positivo y se puede medir restando ambos miembros de la inecuación:

$ABW_{e\text{-}e}(x_i,\ x_n) - ABW_d\ (x_i,\ x_n) =$

$$\sum_{l=i}^{DD-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \left(\sum_{l=i}^{DD-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=DD-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}\right) =$$

$$\sum_{l=i}^{DD-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=i}^{DD-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=DD-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1} =$$

$$\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} - \sum_{l=DD-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1} = \sum_{l=i}^{DD-d-1}\beta_{l,l+1}\,x_{l,l+1}$$

En este caso, el beneficio obtenido en el número de saltos necesarios para la retransmisión de un paquete perdido es:

$n - 1 - i - (n - 1 - (DD - d)) =$

$= n - 1 - i - (n - 1 - DD + d) =$

$= n - 1 - i - n + 1 + DD - d = DD - d - i$

Si $d<DD-i \rightarrow DD-d>i$, $DD-d-i > 0$ se obtiene beneficio y si $d=DD-i \rightarrow DD-d = i$, $DD-d-i = 0$ no se obtiene beneficio ni pérdida, es decir:

$$ABW_d(x_i, x_n) = \sum_{l=i}^{DD-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=DD-d=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1} = ABW_{e2e}(x_i, x_n)$$

Por consiguiente, en el caso de que $d=DD-i$, no hay mejora pero tampoco pérdida, se igualan los recursos consumidos para los casos de recuperación GLRP y extremo a extremo.

Por otro lado, también interesa analizar el problema si existen pérdidas en múltiples nodos intermedios x_{DD}. En este caso, partiendo del ABW total hasta la obtención en x_n del flujo desde el instante inicial de su transmisión para el caso de recuperación *e-e*:

$$ABW_{e2e}(x_i, x_n) = \underbrace{\sum_{l=i}^{i+1-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{i+1-1}\beta_{l,l+1}\,x_{l,l+1}}_{x_{DD}=x_{i+1}:\ (x_i, x_{i+1})+(x_i, x_{i+1})} + \underbrace{\sum_{l=i+1}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}}_{x_{DD}=x_{i+2}:\ (x_{i+1}, x_{i+2})+(x_i, x_{i+2})} +$$

$$+ \underbrace{\sum_{l=i+2}^{i+3-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{i+3-1}\beta_{l,l+1}\,x_{l,l+1}}_{x_{DD}=x_{i+3}:\ (x_{i+2}, x_{i+3})+(x_i, x_{i+3})} + \ \ldots\ + \underbrace{\sum_{l=n-1}^{n-1}\beta_{l,l+1}\,x_{l,l+1} + \sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1}}_{x_{DD}=x_n:\ (x_{n-1}, x_n)+(x_i, x_n)}$$

Así, los recursos consumidos por un paquete reenviado hasta un nodo GLRP intermedio (x_{DD}) se obtiene como la suma de recursos desde x_i hasta x_{DD-1} más los recursos desde el nodo x_{DD-d} hasta el intermedio x_{DD}:

$$ABW_d(x_i, x_n) = \underbrace{\sum_{l=i}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i+1-d}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1}}_{x_{DD} = x_{i+1}:\ (x_i, x_{i+1}) + (x_{i+1-d}, x_{i+1})} + \underbrace{\sum_{l=i+1}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i+2-d}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1}}_{x_{DD} = x_{i+2}:\ (x_{i+1}, x_{i+2}) + (x_{i+2-d}, x_{i+2})} +$$

$$+ \underbrace{\sum_{l=i+2}^{i+3-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i+3-d}^{i+3-1} \beta_{l,l+1}\, x_{l,l+1}}_{x_{DD} = x_{i+3}:\ (x_{i+2}, x_{i+3}) + (x_{i+3-d}, x_{i+3})} + \ ... \ + \underbrace{\sum_{l=n-1}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1}}_{x_{DD} = x_n:\ (x_{n-1}, x_n) + (x_{n-d}, x_n)}$$

Una vez definido $ABW_d(x_i,\ x_n)$, se comprobará que $\Delta_d(x_i,\ x_n) < \Delta_{e\text{-}e}(x_i,\ x_n)$ cuando se produzcan pérdidas en los $n\text{-}i$ últimos nodos:

$$\sum_{l=i}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i+1-d}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i+1}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i+2-d}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + ... +$$

$$+ \sum_{l=n-1}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} \ < \ \sum_{l=i}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1} +$$

$$+ \sum_{l=i+1}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + ... + \sum_{l=n-1}^{n-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

Después de simplificar el término $ABW_{fw}(x_i, x_n) = \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$ en ambos miembros:

$$\underbrace{\sum_{l=i+1-d}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1}}_{d=1} + \sum_{l=i+2-d}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + ... + \sum_{l=n-d}^{n-1} \beta_{l,l+1}\, x_{l,l+1} <$$

$$< \sum_{l=i}^{i+1-1} \beta_{l,l+1}\, x_{l,l+1} + \sum_{l=i}^{i+2-1} \beta_{l,l+1}\, x_{l,l+1} + ... + \sum_{l=i}^{n-1} \beta_{l,l+1}\, x_{l,l+1}$$

Para el nodo x_{i+1}, el único diámetro factible es d=1, de ahí que la ecuación anterior queda de la siguiente forma:

$$\sum_{l=i}^{i+1-1}\beta_{l,l+1}\,x_{l,l+1}+\sum_{l=i+2-d}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots+\sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}<$$

$$<\sum_{l=i}^{i+1-1}\beta_{l,l+1}\,x_{l,l+1}+\sum_{l=i}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots+\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1}$$

$$\underbrace{\sum_{l=i+2-d}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots}_{\text{B}}+\underbrace{\sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}}_{\text{A}}<\underbrace{\sum_{l=i}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots}_{\text{B}}+\underbrace{\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1}}_{\text{A}}$$

Para el término A ya se ha comprobado que $\sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}<\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1}$, con un beneficio en el ABW de $\sum_{l=i}^{n-d-1}\beta_{l,l+1}\,x_{l,l+1}$ y un beneficio de iteraciones de n-d-i. Para el término B se considera el caso n=i+2, n=i+3, ..., n=n-1. Con todo ello queda demostrado que $BW_d\ (x_i,\ x_n) < BW_{e\text{-}e}(x_i,\ x_n)$, cuando el flujo se descarta en los n–i últimos nodos, lo que también implica que el beneficio que obtiene GLRP en el consumo de recursos debe ser positivo y se puede medir restando ambos miembros de la inecuación:

$BW_{e\text{-}e}(x_i,\ x_n) - BW_d\ (x_i,\ x_n)$

$$\sum_{l=i}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots+\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1}-\sum_{l=i+2-d}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots+\sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}=$$

$$\sum_{l=i}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}-\sum_{l=i+2-d}^{i+2-1}\beta_{l,l+1}\,x_{l,l+1}+\sum_{l=i}^{i+3-1}\beta_{l,l+1}\,x_{l,l+1}-\sum_{l=i+3-d}^{i+3-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots+$$

$$+\sum_{l=i}^{n-1}\beta_{l,l+1}\,x_{l,l+1}-\sum_{l=n-d}^{n-1}\beta_{l,l+1}\,x_{l,l+1}$$

$$ABW_{e2e}(x_i,x_n)-ABW_d(x_i,x_n)=$$

$$=\sum_{l=i}^{i+2-d-1}\beta_{l,l+1}\,x_{l,l+1}+\sum_{l=i}^{i+3-d-1}\beta_{l,l+1}\,x_{l,l+1}+\ldots+\sum_{l=i}^{n-d-1}\beta_{l,l+1}\,x_{l,l+1}$$

Por último, el beneficio obtenido con respecto al número de saltos necesarios se puede obtener como: (i+2–d–i)+(i+3–d–i)+...+(n–d–i).

4.4 Análisis probabilístico

En esta sección se llevará a cabo un análisis acerca de la probabilidad de que un paquete perdido se pueda encontrar en el *GBuffer* de algún nodo del *GPlane* en un instante determinado. Este estudio permitiría a un nodo GLRP analizar las posibilidades que hay en un instante dado de recuperar un paquete perdido [7]. Si la probabilidad es baja, el nodo GLRP podría decidir no iniciar el proceso de recuperación GLRP [8].

Como se ha comentado anteriormente, el conjunto de *GBuffer* del *GPlane* se puede considerar como un conjunto de memorias en las que, en algún momento, en cada una de ellas se ha almacenado el paquete perdido pero que, debido al tamaño limitado de las mismas y al flujo continuo de nuevos paquetes del flujo prioritario, el paquete puede haber sido sustituido en algunas de ellas o en todas [9]. En este último caso el paquete no podría retransmitirse localmente y el mensaje *GAck* incluiría una notificación *GAckErr*. En particular, el estudio se centrará en analizar la probabilidad de que el paquete perdido haya sido eliminado ya de todos los *GBuffer*, ya que así el nodo que iniciaría la solicitud de retransmisión local podría tomar la decisión de no comenzar la señalización si la probabilidad de recuperar el paquete es baja [10].

Para comenzar se estudiará la probabilidad básica de que el paquete a solicitar esté entre los n paquetes ya sobrescritos en el *GBuffer* de un único nodo GLRP. Esto se define como la probabilidad $P(GAckErr) = P(X{=}1)$ y representa al caso de imposibilidad de retransmisión local con diámetro $d{=}1$. Si se considera un diámetro mayor, se podrían incluir más nodos a los que hacer la solicitud, existiendo por tanto un mayor número de oportunidades para conseguir una notificación positiva (*GAckOk*). Por ejemplo, para $d{=}2$ y teniendo en cuenta un único flujo que atraviesa ambos nodos, se podría enviar la solicitud a dos nodos GLRP. En este caso se puede calcular la probabilidad de que el paquete perdido tampoco se encuentre almacenado en ninguno de los dos nodos. Esto se representa como la probabilidad $P(GAckErr) = P(X{=}2)$. Sin embargo, ahora se debe tener en cuenta que el número total de paquetes eliminados de los *GBuffer* es $2n$. Este cálculo de probabilidad de no poder recuperar localmente el paquete perdido se puede generalizar para cualquier diámetro: $P(GAckErr) = P(X{=}d)$.

Por otro lado, P es una probabilidad y, como tal, se define como el cociente entre el caso buscado dividido entre los casos posibles. En nuestro caso es el cociente entre el caso en que el paquete a solicitar haya sido sobrescrito en el *GBuffer* de un nodo GLRP dividido entre el número de paquetes almacenados en dicho *GBuffer*, representado como $|GBuffer|$ (el cardinal de *GBuffer*). Si en

lugar del *GBuffer* de un solo nodo se tiene en cuenta todo un *GPlane* de diámetro *d*, en lugar de un solo caso buscado, habrá hasta un máximo de *d* casos buscados, ya que se podrían enviar solicitudes hasta a *d* nodos. Además, en lugar de $|GBuffer|$ casos posibles, se obtienen hasta $\sum_{i=1}^{d}\left|GBuffer_i\right|$ casos posibles.

En esta situación, para calcular la probabilidad p de un valor particular de X (X=d), se considerarán las n sustituciones de paquetes por nuevos paquetes entrantes en los *GBuffer* de los nodos del *GPlane*, de las cuales habrá x sustituciones del paquete buscado (B) y n-x sustituciones del resto de paquetes (R). Esto viene representado por:

$$\underbrace{BBB...B}_{x}\ \underbrace{RRR...\mathrm{R}}_{n-x} = p^x (1-p)^{n-x}$$

Además, para obtener la probabilidad de x casos buscados y n-x casos (el resto) en cualquier orden, se deben sumar las probabilidades de todos los sucesos de forma excluyente. Para ello se combinan los casos B y R en la serie anterior de todas las formas posibles, es decir, se calcula la permutación de n elementos con x y n-x repetidos:

$$\binom{n}{x} = \frac{n!}{x!\,(n-x)!}$$

Entonces, la distribución de probabilidad de un valor particular de X viene dada por:

$$P(X=x) = \binom{n}{x} p^x\,(1-p)^{n-x} = \left(\frac{n!}{x!\,(n-x)!}\right) p^x\,(1-p)^{n-x}, \text{ con } x = 0, 1, 2, ..., \mathrm{n}$$

donde:

n es el número de sustituciones de paquetes

x es el diámetro del *GPlane*.

p es la probabilidad de que el paquete a solicitar haya sido sustituido.

En consecuencia, la probabilidad de obtener un *GAckErr* se calcula como:

$$P(GAckErr)=\left(\frac{\left(\sum_{i=1}^{d}S_i\right)!}{d!\left(\left(\sum_{i=1}^{d}S_i\right)-d\right)!}\right)\left(\frac{d}{\sum_{i=1}^{d}\left|GBuffer_i\right|}\right)^{d}\left(1-\frac{d}{\sum_{i=1}^{d}\left|GBuffer_i\right|}\right)^{\left(\sum_{i=1}^{d}S_i\right)-d}$$

donde:

$d \leq 1, 2, ..., |GPlane|$.

S_i es el número de paquetes que se han sustituido en el *GBuffer* del nodo i desde que se almacenó el paquete a solicitar hasta que se hace la solicitud.

$|GBuffer_i|$: es el número de paquetes almacenados en el *GBuffer* del nodo i.

d: es el número de nodos GLRP en los que se ha almacenado el paquete solicitado.

Cuanto mayor sea el valor de $|GBuffer_i|$ más difícil será que el paquete que se va a solicitar se haya sustituido ya. En general, si en el túnel LSP hay d nodos con capacidad GLRP, la probabilidad de que se sustituya el paquete a solicitar ante la llegada de un nuevo paquete a cada uno de los nodos es:

$$p = \frac{d}{\sum_{i=1}^{d}\left|GBuffer_i\right|}$$

Es decir, d casos buscados dividido por $\sum_{i=1}^{d}\left|GBuffer_i\right|$ casos posibles.

Así mismo, el valor de S en un instante t y en un nodo i viene dado por:

$S_i = PacketRate_i \cdot t - |GBuffer_i|$, donde $PacketRate_i$ es la velocidad de llegada de nuevos paquetes del mismo *GLevel* al *GBuffer* del nodo i por unidad de tiempo.

Finalmente se puede analizar la relación existente entre la probabilidad de encontrar un paquete y el número de nodos disponibles en el *GPlane* a los que poder hacer dicha solicitud. Por ejemplo, para un caso simple en el que se pueden almacenar 100 paquetes de un *GLevel* en los *GBuffer*, se puede comprobar a continuación cómo disminuye la probabilidad de obtener la respuesta *GAckErr* a medida que aumenta el número de nodos a los que poder enviar el mensaje *GReq*:

$d = 1$

$S = d \cdot 20 = 20$

$$P(GAckErr) = P(X = 1) = \left(\frac{20!}{1!\,(20-1)!} \right) \cdot \left(\frac{1}{100} \right)^{1} \cdot \left(1 - \frac{1}{100} \right)^{20-1} = 20 \cdot 0{,}01 \cdot (0{,}99)^{19} = 0{,}1652$$

$d = 2$

$S = d \cdot 20 = 40$

$$P(GAckErr) = \left(\frac{40!}{2!\,(40-2)!} \right) \cdot \left(\frac{2}{200} \right)^{2} \cdot \left(1 - \frac{2}{200} \right)^{40-2} = 780 \cdot 0{,}0001 \cdot (0{,}99)^{38} = 0{,}0532$$

$d = 3$

$S = d \cdot 20 = 60$

$$P(GAckErr) = \left(\frac{60!}{3!\,(60-3)!} \right) \cdot \left(\frac{3}{300} \right)^{3} \cdot \left(1 - \frac{3}{300} \right)^{60-3} = 34220 \cdot 0{,}000001 \cdot (0{,}99)^{57} = 0{,}0193$$

$d = 4$

$S = d \cdot 20 = 80$

$$P(GAckErr) = \left(\frac{80!}{4!\,(80-4)!} \right) \cdot \left(\frac{4}{400} \right)^{4} \cdot \left(1 - \frac{4}{400} \right)^{80-4} = 1581580 \cdot 10^{-8} \cdot (0{,}99)^{76} = 0{,}0074$$

Con los anteriores ejemplos se ha comprobado que a medida que aumenta el diámetro del *GPlane*, disminuye la probabilidad de no poder encontrar el paquete GLRP en ningún nodo. Así pues debe aumentar el suceso contrario, es decir, la probabilidad de encontrarlo en al menos uno de los nodos del *GPlane*. Para comprobarlo se calculará ahora la probabilidad de obtener un *GAckOk* , lo cual denota que el paquete se encuentra en al menos uno de los nodos del *GPlane*. El objetivo sería ahora calcular $P(X<d)$, que representa la probabilidad de que el paquete GLRP buscado no haya sido eliminado ya de todos los *GBuffer*. Por ejemplo, $P(X=d\text{-}1)$ es la probabilidad de que el paquete se haya eliminado de todas los nodos excepto de uno, $P(X=d\text{-}2)$ indica que el paquete podría ser recuperado localmente de dos nodos que aún lo contienen, $P(X=d\text{-}d)$ indica que el paquete no se ha eliminado aún de ninguna memoria y podrían establecerse, por tanto, d planos GLRP para solicitar su retransmisión local. En resumen, $P(GAckOk) = P(X < d) = P(X=d\text{-}1) + P(X=d\text{-}2) + P(X=d\text{-}3) + ... + P(X=d\text{-}d)$:

$$P(GAckOk) = P(X < d) = \sum_{j=1}^{d} P(X = d - j) =$$

$$= \sum_{j=1}^{d} \left(\left(\frac{\left(\sum_{i=1}^{d} S_i \right)!}{(d-j)! \left(\left(\sum_{i=1}^{d} S_i \right) - (d-j) \right)!} \right) \cdot \left(\frac{d}{\sum_{i=1}^{d} \left| GBuffer_i \right|} \right)^{d-j} \cdot \left(1 - \frac{d}{\sum_{i=1}^{d} \left| GBuffer_i \right|} \right)^{\left(\sum_{i=1}^{d} S_i \right) - (d-j)} \right)$$

En los siguientes ejemplos se puede comprobar que a medida que aumenta el *diámetro* del *GPlane* o, lo que es lo mismo, el número de nodos a los que poder enviar un mensaje *GReq*, también aumenta la probabilidad de obtener la respuesta *GAckOk* por encontrar el paquete en alguno de los nodos:

$d = 1$

$S = d \cdot 20 = 20$

$$P(GAckOk) = P(X < 1) = \sum_{i=1}^{1} P(X = 1 - i) = P(X = 0)$$

$$P(GAckOk) = \left(\frac{20!}{0!\,(20-0)!} \right) \cdot \left(\frac{1}{100} \right)^{0} \cdot \left(1 - \frac{1}{100} \right)^{20-0} = 1 \cdot (0{,}01)^{0} \cdot (0{,}99)^{20} = 0{,}818$$

$d = 2$

$S = d \cdot 20 = 40$

$$P(GAckOk) = P(X < 2) = \sum_{i=1}^{2} P(X = 2 - i) = P(X = 1) + P(X = 0)$$

$$P(GAckOk) = \left(\frac{40!}{1!\,(40-1)!} \right) \cdot \left(\frac{2}{200} \right)^{1} \cdot \left(1 - \frac{2}{200} \right)^{40-1} + \left(\frac{40!}{0!\,(40-0)!} \right) \cdot \left(\frac{2}{200} \right)^{0} \cdot \left(1 - \frac{2}{200} \right)^{40-0} =$$

$$= 40 \cdot 0{,}01 \cdot (0{,}99)^{39} + 1 \cdot (0{,}01)^{0} \cdot (0{,}99)^{40} = 0{,}9393$$

$d = 3$

$S = d \cdot 20 = 60$

$$P(GAckOk) = P(X < 3) = \sum_{i=1}^{3} P(X = 3 - i) = P(X = 2) + P(X = 1) + P(X = 0)$$

$$P(GAckOk) = \left(\frac{60!}{2!\,(60-2)!} \right) \cdot \left(\frac{3}{300} \right)^{2} \cdot \left(1 - \frac{3}{300} \right)^{60-2} + \left(\frac{60!}{1!\,(60-1)!} \right) \cdot \left(\frac{3}{300} \right)^{1} \cdot \left(1 - \frac{3}{300} \right)^{60-1} +$$

$$+ \left(\frac{60!}{0!\,(60-0)!} \right) \cdot \left(\frac{3}{300} \right)^{0} \cdot \left(1 - \frac{3}{300} \right)^{60-0} = 1770 \cdot 0{,}0001 \cdot (0{,}99)^{58} + 60 \cdot 0{,}01 \cdot (0{,}99)^{59} + 0{,}99^{60} =$$

$$= 0{,}9776$$

$d = 4$

$S = d \cdot 20 = 80$

$$P(GAckOk) = P(X < 4) = \sum_{i=1}^{4} P(X = 4 - i) =$$

$$= P(X = 3) + P(X = 2) + P(X = 1) + P(X = 0) =$$

$$= \left(\frac{80!}{3!\,(80-3)!}\right) \cdot \left(\frac{4}{400}\right)^3 \cdot \left(1 - \frac{4}{400}\right)^{80-3} + \left(\frac{80!}{2!\,(80-2)!}\right) \cdot \left(\frac{4}{400}\right)^2 \cdot \left(1 - \frac{4}{400}\right)^{80-2} +$$

$$+ \left(\frac{80!}{1!\,(80-1)!}\right) \cdot \left(\frac{4}{400}\right)^1 \cdot \left(1 - \frac{4}{400}\right)^{80-1} + \left(\frac{80!}{0!\,(80-0)!}\right) \cdot \left(\frac{4}{400}\right)^0 \cdot \left(1 - \frac{4}{400}\right)^{80-0} =$$

$$= 82160 \cdot (0{,}01)^3 \cdot (0{,}99)^{77} + 3160 \cdot (0{,}01)^2 \cdot (0{,}99)^{78} + 80 \cdot 0{,}01 \cdot (0{,}99)^{79} + (0{,}99)^{80} = 0{,}9913$$

En la Figura 4-8 se observa que a medida que se aumenta el diámetro del *GPlane*, también aumenta la probabilidad de obtener la notificación *GAckOk*. De igual forma, también mejora al aumentar el tamaño del *GBuffer*. Al aumentar el diámetro, es decir, para $d > 1$, se tiene más de una oportunidad para encontrar el paquete solicitado. Como ya se ha comentado, en estos casos la probabilidad de que el paquete se localice en al menos uno de los nodos del *GPlane* implica la suma de las probabilidades.

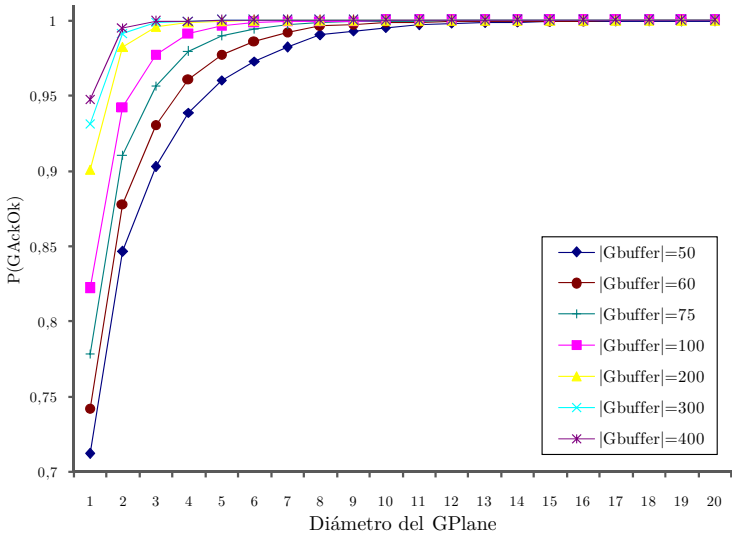

Figura 4-8. Probabilidad de obtener *GAckOk* en función del diámetro del *GPlane* y del tamaño del *GBuffer*

Para el caso d=2 hay dos oportunidades para conseguir la notificación *GAckOk*, denotando que el paquete se encuentra en ambos nodos del *GPlane* ($P(X$=d-2)), o en uno de ellos $P(X$=d-1)). Así, la probabilidad buscada es la suma de ambas: $P(X<d) = P(X$=d-2) + $P(X$=d-1). En la Figura 4-9 se muestran las funciones $P(X$=d-2), $P(X$=d-1) y $P(X<d)$ para d=2 y con diferentes tamaños de *GBuffer*.

Para el caso $P(X$=d-2) = $P(X$=0) se observa que a medida que llegan nuevos paquetes al nodo, la probabilidad de que el paquete a solicitar se encuentre en ambos nodos es cada vez menor.

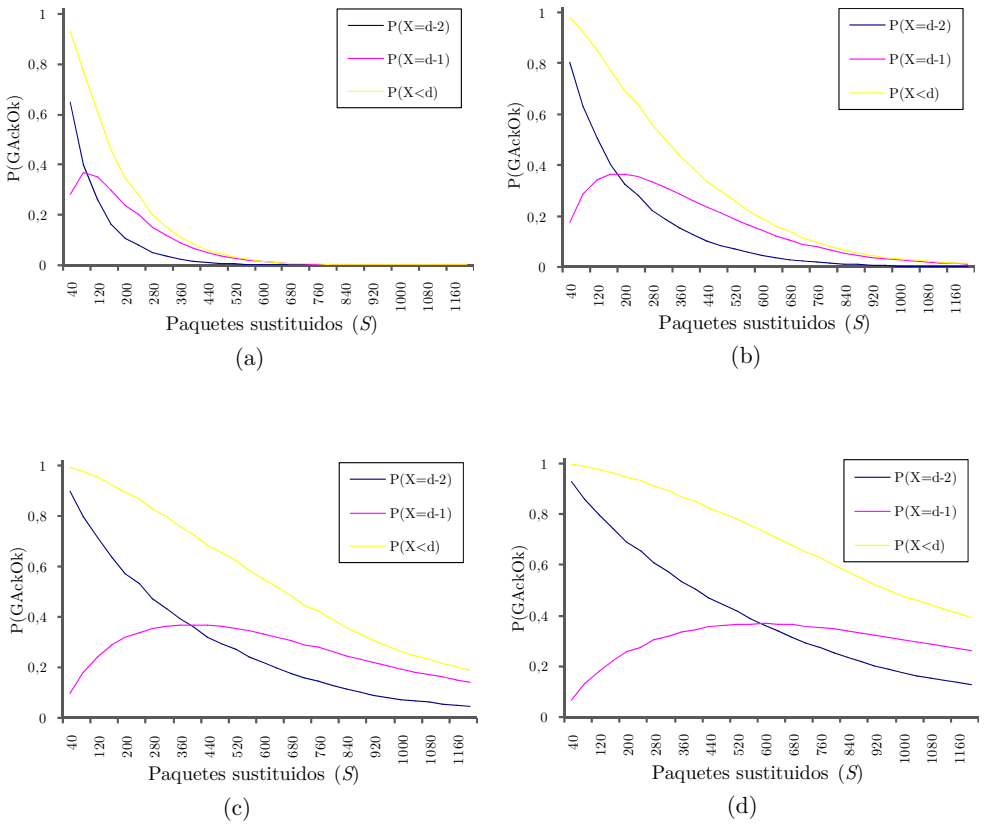

Figura 4-9. Probabilidad de obtener *GAckOk* en función de S, para d=2:
(a) para |*GBuffer*|=100; (b) para |*GBuffer*|=200;
(c) para |*GBuffer*|=400; (d) para |*GBuffer*|=600

Sin embargo, para el caso $P(X{=}d\text{-}1) = P(X{=}1)$ se observa que la función no es decreciente en todos sus puntos, sino que se distingue una primera parte creciente y una segunda en la que la curva disminuye. De hecho, al principio la probabilidad de que el paquete a buscar se haya eliminado de uno solo de los nodos es cada vez mayor. Este es el motivo por el cual al principio aumente la probabilidad de que el paquete esté almacenado en sólo uno de los nodos. Este hecho se produce hasta un punto máximo en el que a partir de ahí lo que aumenta es la probabilidad de que el paquete sea sustituido en ambos nodos. Por ello, a partir de este punto comienza a decrecer la probabilidad de que el paquete esté almacenado en sólo uno de los nodos. En la Figura 4-10 se muestra esta probabilidad con d=4 para diferentes tamaños del *GBuffer*. En la Figura 4-11 se muestra el mismo estudio para d=8.

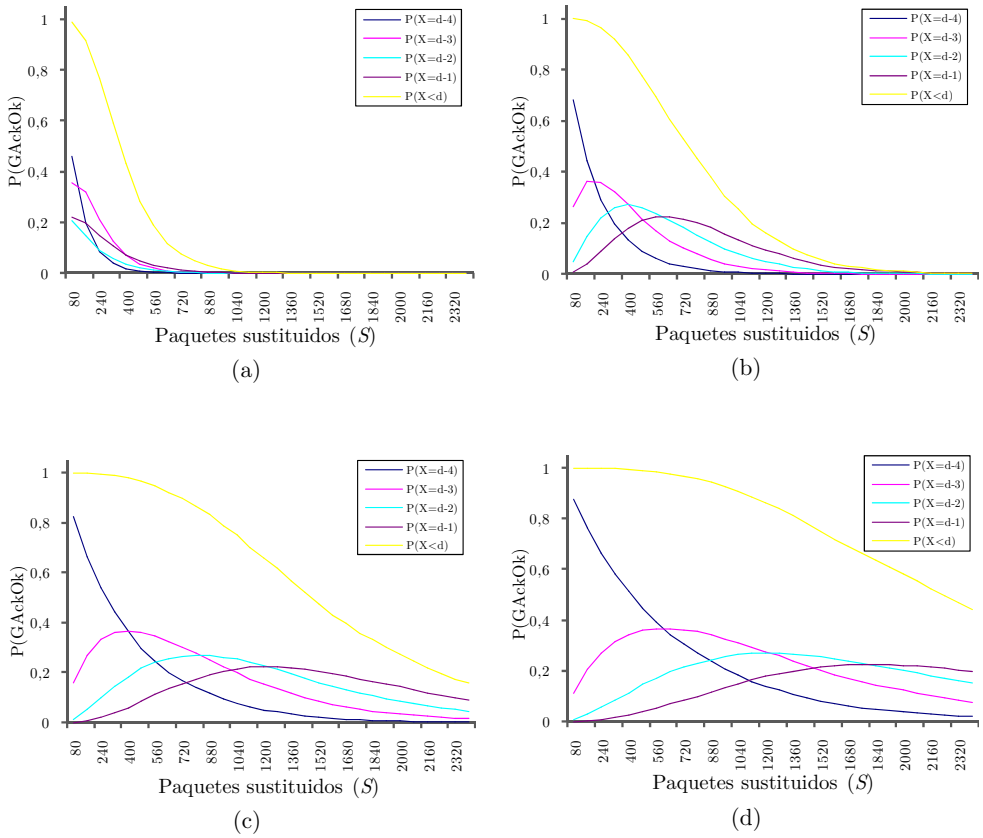

Figura 4-10. Probabilidad de obtener *GAckOk* en función de S, para d=4:
(a) para |*GBuffer*|=100; (b) para |*GBuffer*|=200;
(c) para |*GBuffer*|=400; (d) para |*GBuffer*|=600

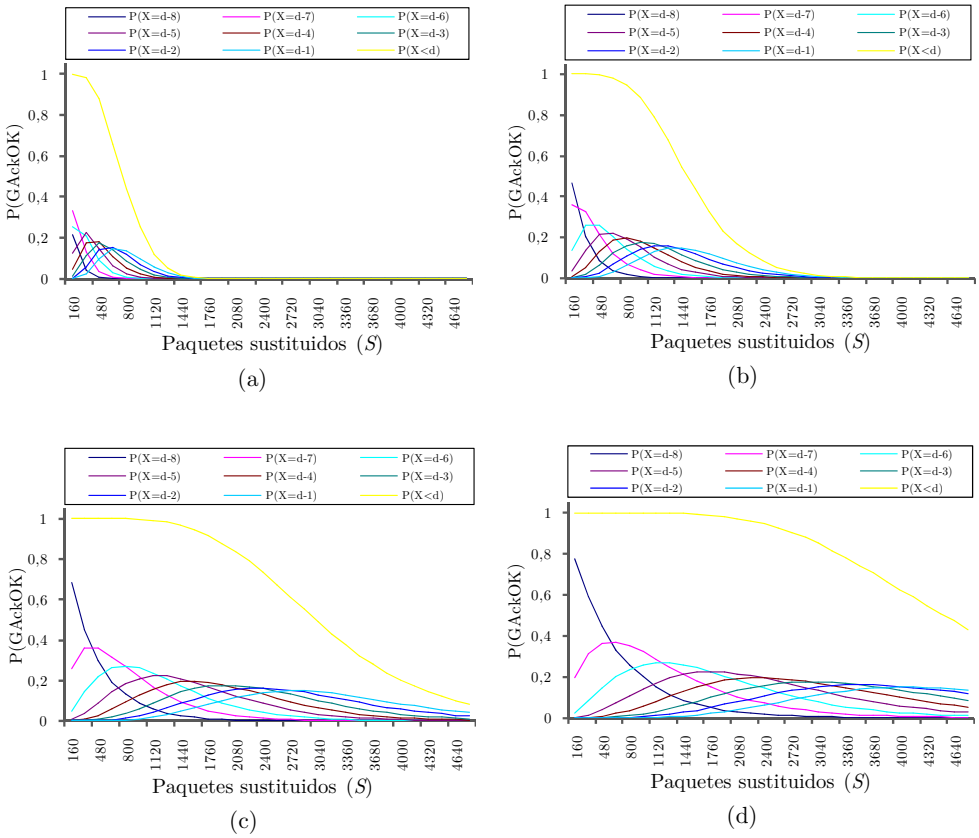

Figura 4-11. Probabilidad de obtener *GAckOk* en función de *S*, para *d*=8:
(a) para |*GBuffer*|=100; (b) para |*GBuffer*|=200;
(c) para |*GBuffer*|=400; (d) para |*GBuffer*|=600

En las gráficas de la Figura 4-12 a la Figura 4-15 se muestran comparativas de la probabilidad total $P(X<d)$ para diferentes diámetros de recuperación local, desde d=1 hasta d=8, respectivamente. En este sentido, cuando el diámetro aumenta también mejora la probabilidad de obtener una notificación *GAckOk* debido a que el paquete a solicitar se encuentre en al menos uno de los nodos del *GPlane*.

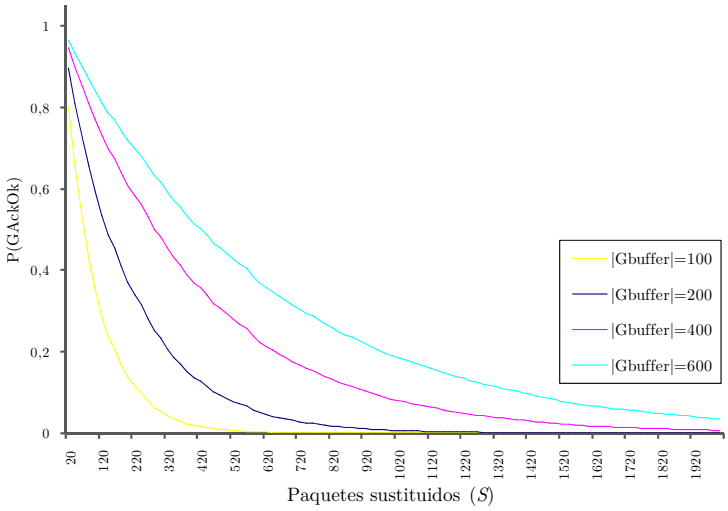

Figura 4-12. Probabilidad de obtener *GAckOk* en función de *S*, para *d*=1

Como era de esperar, se puede apreciar en cada figura que la probabilidad disminuye en función de la llegada de nuevos paquetes del mismo *GLevel* al *GBuffer*, debido a las sustituciones. Sin embargo, ese decremento está relacionado con el tamaño del *GBuffer*.

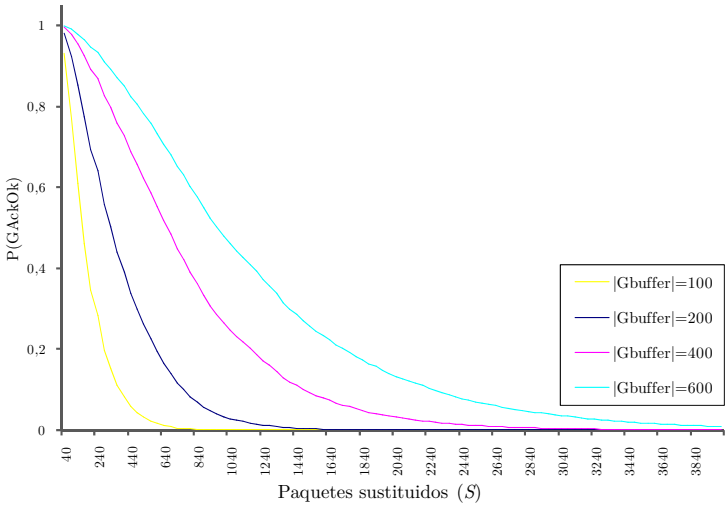

Figura 4-13. Probabilidad de obtener *GAckOk* en función de *S*, para *d*=2

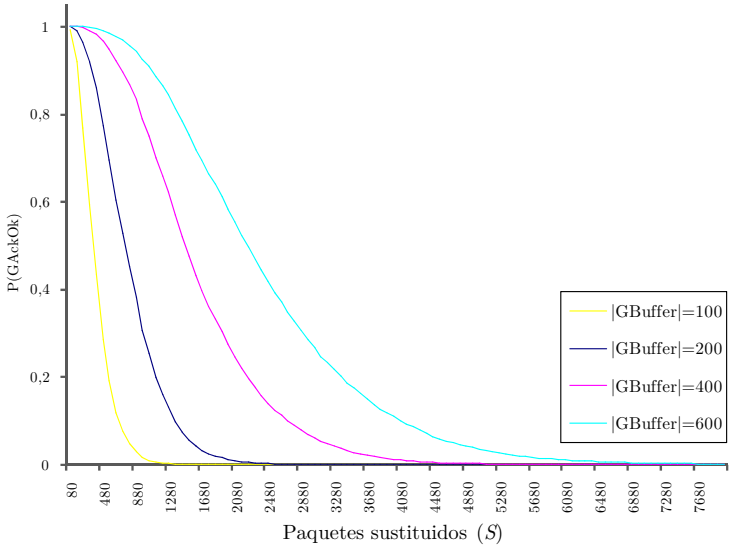

Figura 4-14. Probabilidad de obtener *GAckOk* en función de *S*, para *d*=4

Concretamente, al aumentar su tamaño también mejora $P(GLRPAckOk)$, ya que las funciones que representan los valores mayores de *GBuffer* tienen menor pendiente.

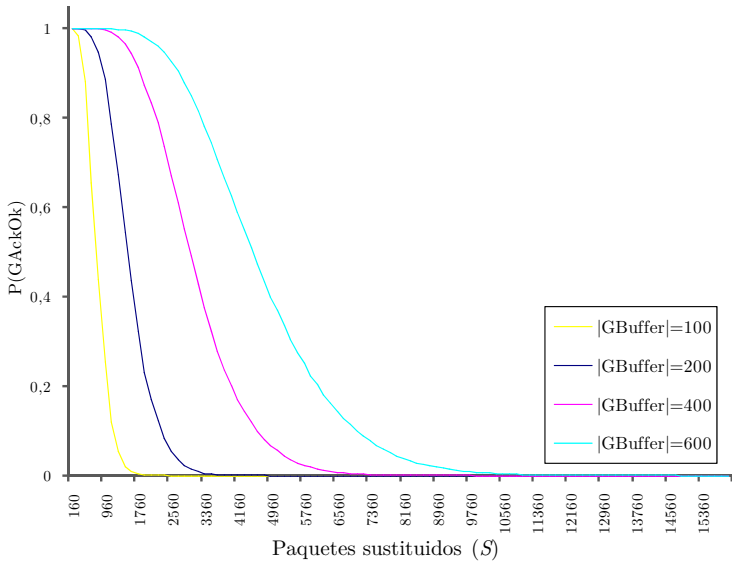

Figura 4-15. Probabilidad de obtener *GAckOk* en función de *S*, para *d*=8

Por otro lado, en las gráficas de la Figura 4-16 a la Figura 4-19 se muestran comparativas de $P(GLRPAckOk)$ en función de la velocidad de llegada de nuevos paquetes prioritarios a los nodos del *GPlane*, para ratios de 350, 700, 1400 y 2800 paquetes por segundo, respectivamente. Al igual que anteriormente, se tendrán en cuenta diferentes diámetros del *GPlane*, así como varios tamaños de *GBuffer*.

En este caso, a medida que aumenta la velocidad de llegada de nuevos paquetes prioritarios al *GPlane*, la probabilidad de encontrar el paquete a solicitar disminuye. Sin embargo, se sigue observando que a medida que se aumenta el tamaño del *GBuffer* la probabilidad es mayor.

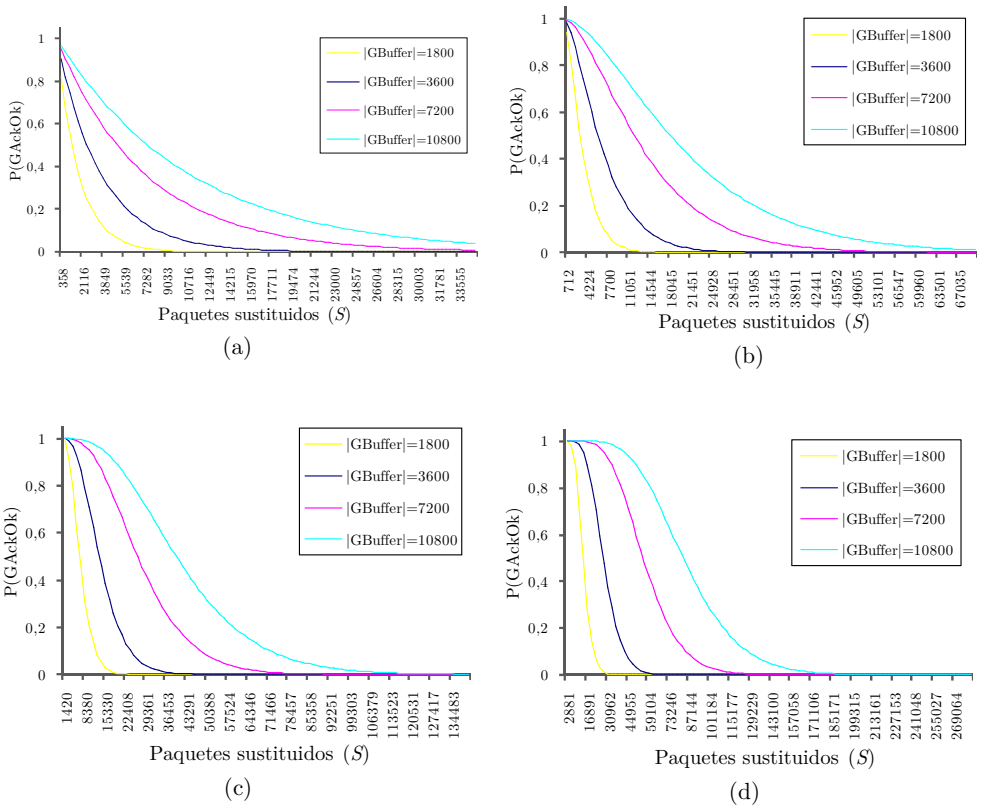

Figura 4-16. Probabilidad de obtener *GAckOk* en función de *S* para una ratio de llegada de nuevos paquetes de 350p/s: (a) para *d*=1; (b) para *d*=2; (c) para *d*=4; (d) para *d*=8

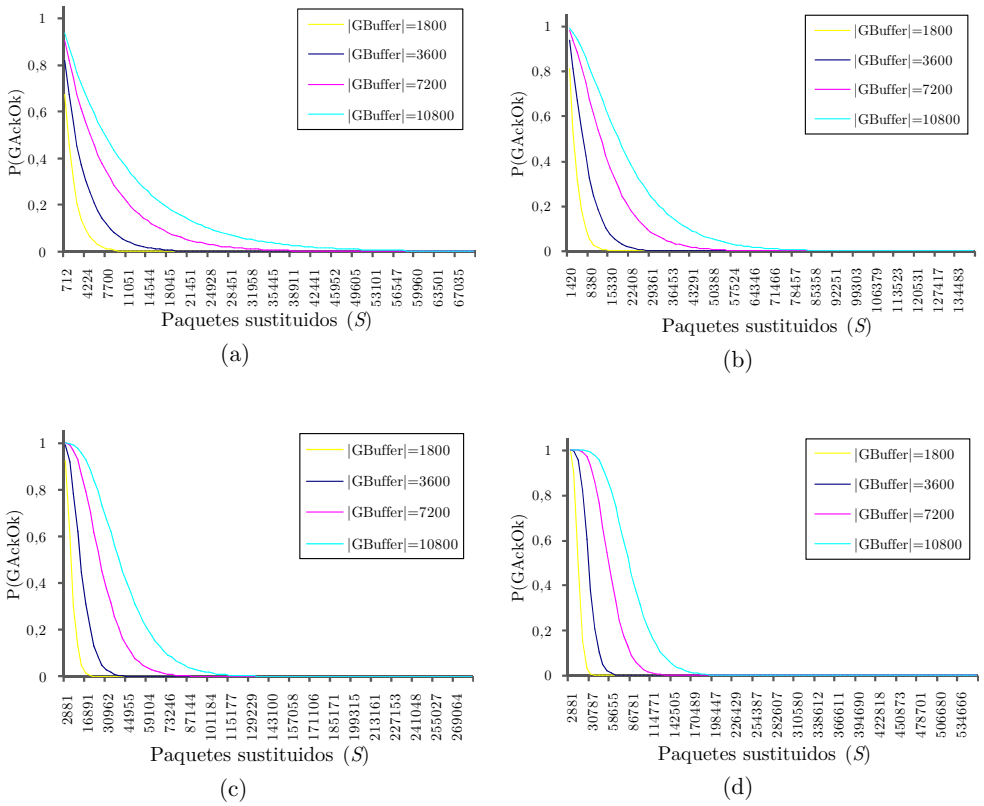

Figura 4-17. Probabilidad de obtener *GAckOk* en función de S para una ratio de llegada de nuevos paquetes de 700p/s: (a) para d=1; (b) para d=2; (c) para d=4; (d) para d=8

También para mayores diámetros de recuperación local la probabilidad es mayor, ya que si d es mayor esto implica que más nodos tienen almacenado el paquete, por tanto la probabilidad de encontrarlo es mayor.

No obstante, hasta ahora sólo se ha llevado a cabo un análisis teniendo en cuenta el número de paquetes que se han debido sustituir en los *GBuffer* de los nodos del *GPlane*, debido al tamaño limitado de éstos. Además, es importante destacar que el estudio se ha realizado para todo el tiempo que dura la transmisión del mensaje.

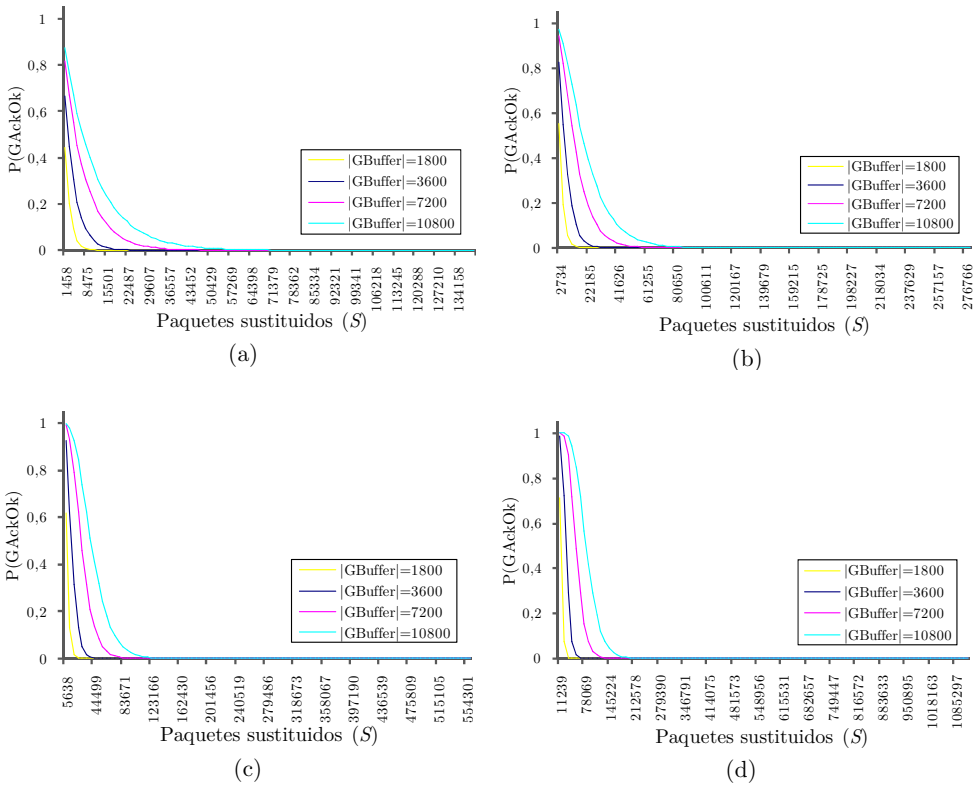

Figura 4-18. Probabilidad de obtener *GAckOk* en función de S para una ratio de llegada de nuevos paquetes de 1400p/s: (a) para d=1; (b) para d=2; (c) para d=4; (d) para d=8

En cierto modo, el intervalo en que resulta útil que esté almacenado un paquete prioritario en el *GBuffer* de un nodo particular es mucho más pequeño que el analizado hasta ahora. Este tiempo de vida útil es el *Round Trip Time* considerado entre el nodo que potencialmente puede retransmitir un paquete localmente y el nodo que podría solicitar la retransmisión local del mismo.

Es decir, es la suma del tiempo que transcurre desde que el paquete pasa por el nodo GLRP que lo almacena hasta que el paquete llega al nodo en que se descarta, más el tiempo que transcurre desde que sale el mensaje *GReq* de este nodo hasta que llega al nodo que lo almacenó.

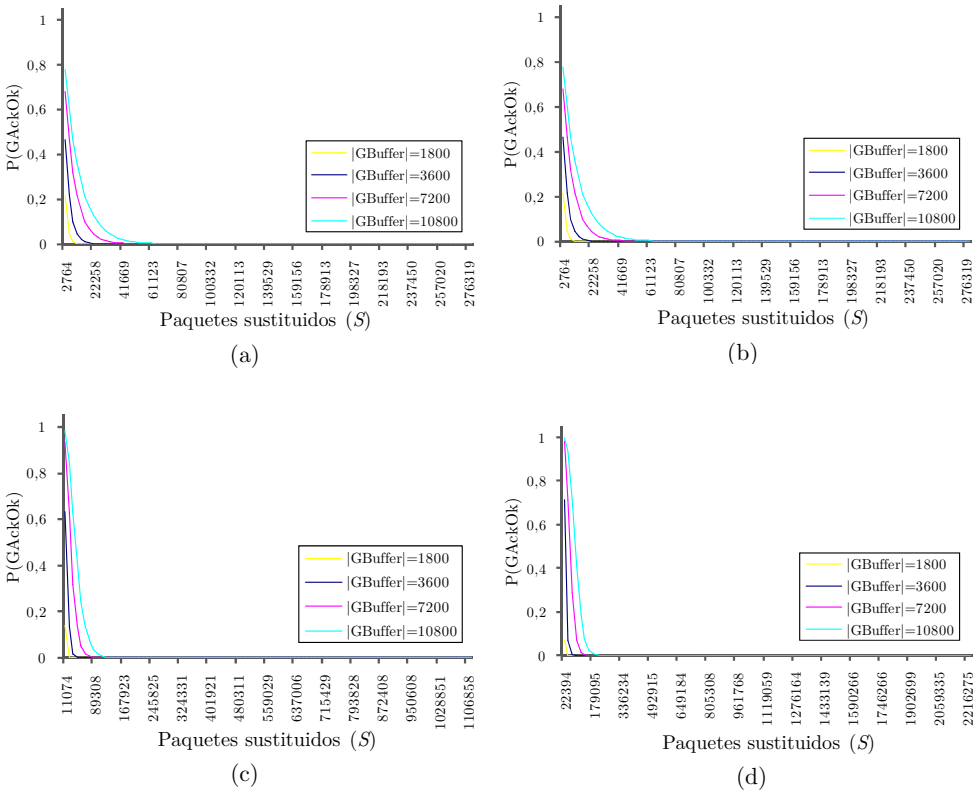

Figura 4-19. Probabilidad de obtener *GAckOk* en función de *S* para una ratio de llegada de nuevos paquetes de 2800p/s: (a) para *d*=1; (b) para *d*=2; (c) para *d*=4; (d) para *d*=8

En las gráficas de la Figura 4-20 a la Figura 4-23, respectivamente, se hace ahora un análisis de $P(GAckOk)$ en función del tiempo transcurrido desde que un paquete fue almacenado en el *GBuffer* de un nodo GLRP. El análisis consiste en una comparativa entre los diferentes diámetros de recuperación local, para varios tamaños de *GBuffer*, y teniendo en cuenta distintas ratios de llegada de nuevos paquetes.

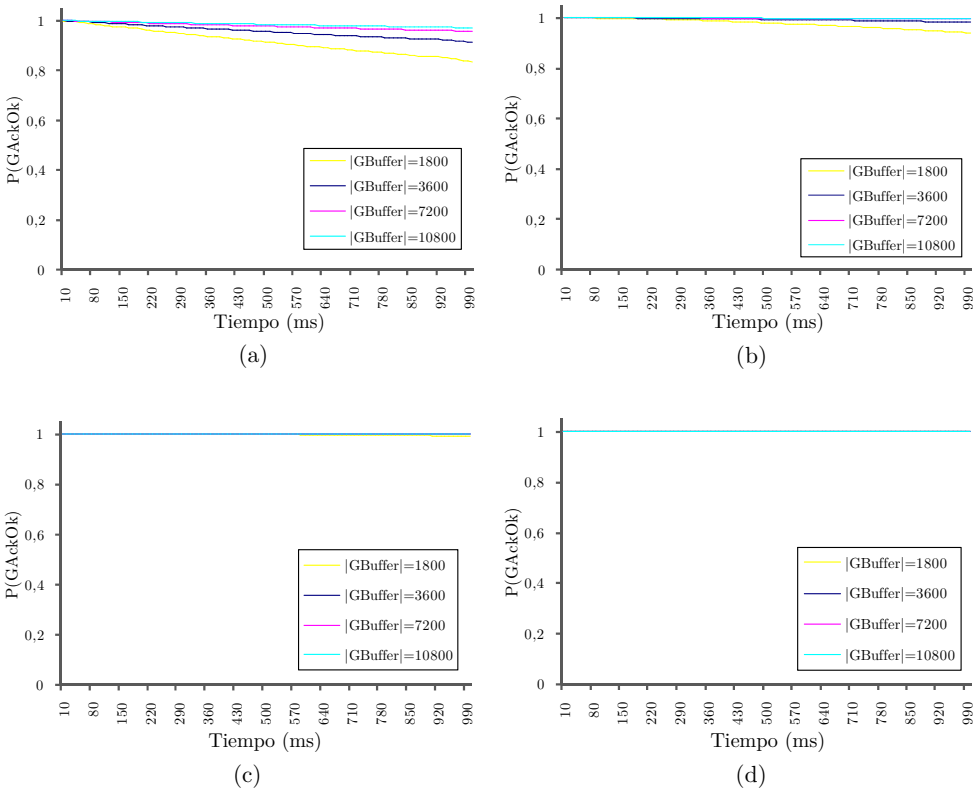

Figura 4-20. Probabilidad de obtener *GAckOk* en función del tiempo para una ratio de llegada de nuevos paquetes de 350p/s: (a) para d=1; (b) para d=2; (c) para d=4; (d) para d=8

El análisis se hace para un tiempo máximo de 1000ms desde que el paquete fue almacenado en un *GBuffer* y se considera, como anteriormente, ratios de llegada de nuevos paquetes de 350, 700, 1400 y 2800 paquetes por segundo, correspondientes a las.

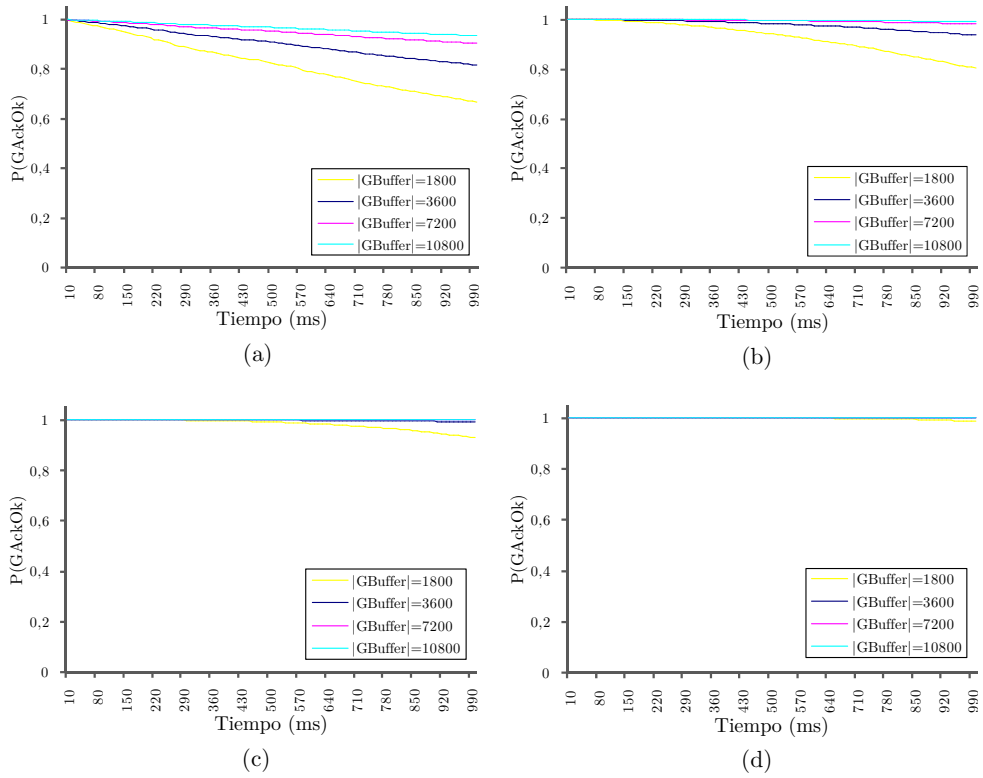

Figura 4-21. Probabilidad de obtener *GAckOk* en función del tiempo para una ratio de llegada de nuevos paquetes de 700p/s: (a) para d=1; (b) para d=2; (c) para d=4; (d) para d=8

Al igual que anteriormente, la probabilidad de obtener la notificación *GAckOk* es mayor a medida que se aumenta el tamaño del *GBuffer*. También para mayores valores del diámetro la probabilidad es mejor, ya que un mayor diámetro implica que más nodos tienen almacenado el paquete, por tanto, dentro de los 1000ms del estudio, la probabilidad de encontrarlo es mayor.

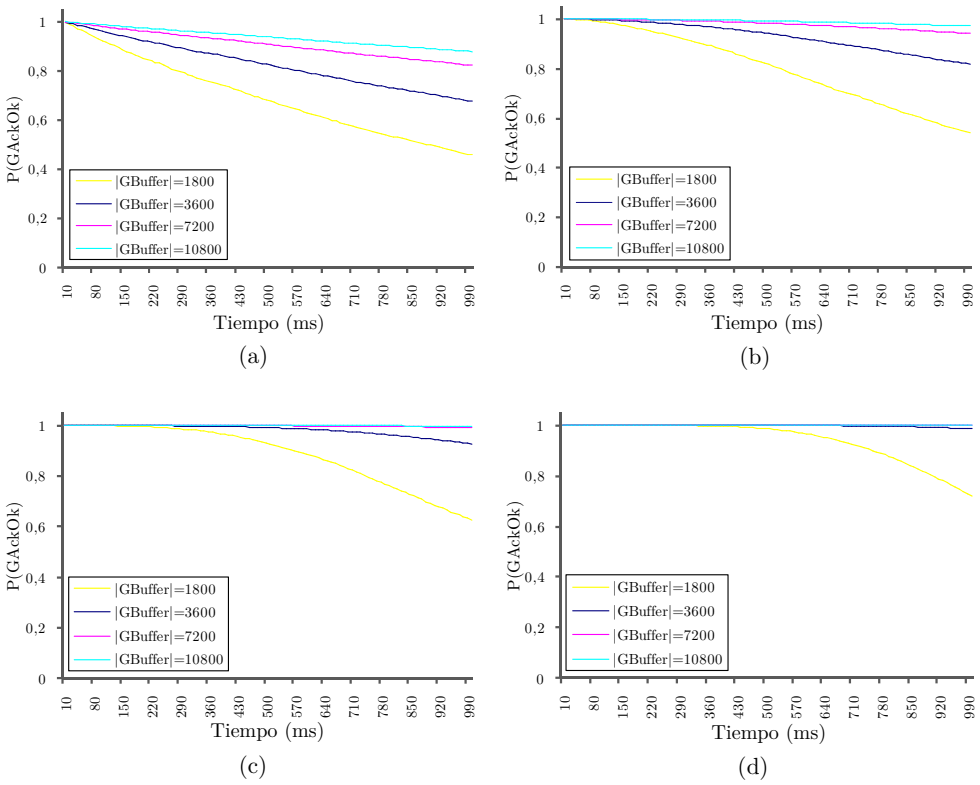

Figura 4-22. Probabilidad de obtener *GAckOk* en función del tiempo para una ratio de llegada de nuevos paquetes de 1400p/s: (a) para d=1; (b) para d=2; (c) para d=4; (d) para d=8

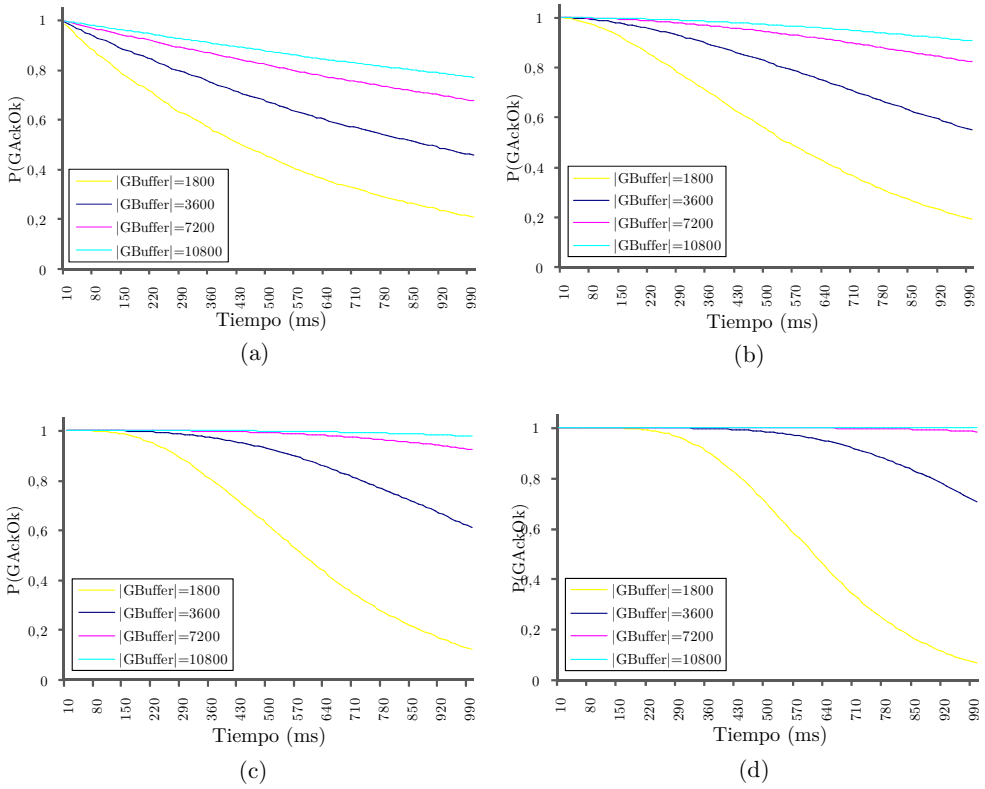

Figura 4-23. Probabilidad de obtener *GAckOk* en función del tiempo para una ratio de llegada de nuevos paquetes de 2800p/s: (a) para *d*=1; (b) para *d*=2; (c) para *d*=4; (d) para *d*=8

4.5 Referencias del capítulo

[1] C. Wang, B. Li and Y. Hu, "Upstream congestion control in wireless sensor networks through cross-layer optimization," IEEE Journal on Selected Areas in Communications, vol 25, nº 4. Pag 786 - 795. 2007. [Online]: http://dx.doi.org/10.1109/JSAC.2007.070514 (último acceso julio 2014).

[2] M.S. Akbar, M.A. Jinnah, S.Z. Ahmad and M.A. Qadir, "Adaptive congestion control mechanism of TCP flows for performance optimization in mobile heterogeneous wireless networks," 2nd International Conference on Computer, Control and Communication. Pag. 1 - 6. 2009. [Online]: http://dx.doi.org/10.1109/IC4.2009.4909164 (último acceso julio 2014).

[3] Y. Zhang, F. Wang and Miao Li, "Research on Reliability Optimization Method for Mesh Network Communication Based on Node Congestion Degree," International Conference on Computational Intelligence and Software Engineering (CiSE). Pag. 1 - 4. 2010. [Online]: http://dx.doi.org/10.1109/CISE.2010.5676936 (último acceso julio 2014).

[4] M. Zhang, Bin Liu and B. Zhang, "Multi-Commodity Flow Traffic Engineering with Hybrid MPLS/OSPF Routing," IEEE Global Telecommunications Conference (GLOBECOM). Pag. 1 - 6. 2009. [Online]: http://dx.doi.org/10.1109/GLOCOM.2009.5426135 (último acceso julio 2014).

[5] S. Veni, G.M.K. Nawaz and P. Praba, "Performance analysis of network traffic behavior in conventional network over MPLS," IEEE International Conference on Communication Control and Computing Technologies (ICCCCT). Pag. 222 - 226. 2010. [Online]: http://dx.doi.org/10.1109/ICCCCT.2010.5670555 (último acceso julio 2014).

[6] Y. Jiangzhou and L. Zengji, "Resource allocation and admission control based on flow congestion probability in MPLS networks," 11th International Conference on Advanced Communication Technology (ICACT), vol 1. Pag. 694 - 697. 2009.

[7] D. Zhang and D. Ionescu, "Measurement and Control of Packet Loss Probability for MPLS VPN Services," IEEE Transactions on Instrumentation and Measurement, vol 55, nº 5. Pag 1587 - 1598. 2006. [Online]: http://dx.doi.org/10.1109/TIM.2006.881583 (último acceso julio 2014).

[8] M. Arumaithurai, R. Geib, R. Rex and Xiaoming Fu, "Pre-congestion notification-based flow management in MPLS-based DiffServ networks," IEEE 28th International Performance Computing and Comm. Conference (IPCCC). Pag. 57 - 64. 2009. [Online]: http://dx.doi.org/10.1109/PCCC.2009.5403827 (último acceso julio 2014).

[9] D. Zhang and D. Ionescu, "Providing Guaranteed Packet Loss Probability Service in IP/MPLS-Based Networks," IEEE International Conference on Communications. Pag. 5772 - 5776. 2008. [Online]: http://dx.doi.org/10.1109/ICC.2008.1080 (último acceso julio 2014).

[10] A. Alwehaibi, M. Kadoch and A. Elhakeem, "Packet loss probability for DiffServ over heterogeneous MPLS multicast networks: a simulation study," Canadian Conference on Electrical and Computer Engineering, Vol 4. Pag. 2209 - 2212. 2004. [Online]: http://dx.doi.org/10.1109/CCECE.2004.1347683 (último acceso julio 2014).

Capítulo 5. Evaluación de resultados

La ciencia más útil es aquella cuyo fruto es el más comunicable.
Leonardo Da Vinci

En el anterior capítulo se llevó a cabo un análisis estadístico y probabilístico de la propuesta, con el objetivo de encontrar las ecuaciones de mejora de retardo y consumo de recursos de la red mediante GLRP, con respecto a los esquemas tradicionales de control de pérdidas extremo a extremo. En este capítulo se presentan las herramientas desarrolladas para llevar a cabo las simulaciones que permiten determinar el comportamiento de GLRP en función de diferentes parámetros y condiciones. También se analizan las pruebas realizadas y los resultados obtenidos.

5.1 Simulador *OpenSimMPLS*

OpenSimMPLS es una aplicación de simulación de redes multilingüe y portable que ha sido desarrollada en el contexto del Grupo de investigación de Ingeniería Telemática Aplicada y Comunicaciones Avanzadas (GÍTACA). Inicialmente fue empleado con fines de innovación docente, aunque también ha servido como plataforma para la prueba empírica de las conclusiones derivadas de diversas investigaciones [1]. Se trata de una herramienta que ha sido descargada más de 53.000 veces desde 140 países distintos y que contempla los aspectos fundamentales de configuración y operación de un dominio MPLS, incluyendo también compatibilidad con dominios que soporten Garantía de Servicio mediante técnicas activas. Se trata de un simulador que permite dotar a flujos prioritarios de garantías de servicio, utilizando para ello MPLS como solución de red/enlace sobre IP, LDP como solución de protocolo de señalización de LSPs en el dominio MPLS, y GPSRP (*GoS PDU Store and Retransmit Protocol*) como

protocolo propio de recuperación de paquetes prioritarios. Su finalidad es la simulación de escenarios completos basados en redes MPLS con soporte de garantía de servicio mediante técnicas activas, con la que se pueden recrear dichos escenarios y comprobar su comportamiento. Es una aplicación desarrollada con *Java*, por lo que es muy portable. En la Figura 5-1 se muestra una captura de ejemplo de un escenario de simulación de *OpenSimMPLS*.

OpenSimMPLS se ha liberado con licencia *GPL 3.0*, por lo que se permite la modificación de su código. De hecho, lo hemos extendido para dar también soporte a la funcionalidad de Capa de Transporte, a la señalización RSVP-TE y a nuestra propuesta GLRP, dando lugar a *OpenSimMPLSv2*, la cual está así más en el contexto de la presente tesis que la versión original. Principalmente se ha empleado en la docencia de asignaturas de redes o comunicaciones, ya que permite al estudiante la generación visual de escenarios de una forma muy flexible, así como la simulación interactiva de dominios de red MPLS con soporte de control de errores mediante GLRP. Esta interactividad permite, al igual que en la versión original, la congestión manual de nodos, fallos de enlace o nodo, análisis de estadísticas de funcionamiento, etc., pero permitiendo ahora la señalización de LSP mediante RSVP-TE, la configuración del *GPlane* GLRP o la comparación de las estadísticas de funcionamiento con el rendimiento que ofrecería la Capa de Transporte.

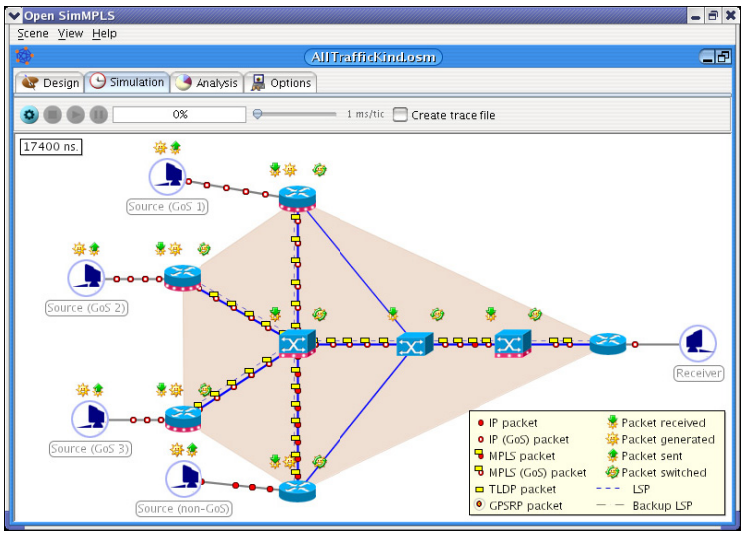

Figura 5-1. Ejemplo de escenario de simulación en *OpenSimMPLS*

En particular, se ha implementado TCP como protocolo de transporte, para dar fiabilidad a las transmisiones. Proporciona retransmisiones extremo a extremo de paquetes perdidos, por lo que ha sido necesario implementar la ventana de transmisión, el RTO (*Retransmission TimeOut*), y los algoritmos comunes *Slow Start*, *Congestion Avoidance*, y *Fast Retransmit*. En el emisor se llevarán a cabo los controles de flujo, de congestión y de retransmisiones, mientras que el receptor lleva a cabo el control de errores, comprobando además que el número de secuencia recibido sea el esperado.

Por otro lado, se ha implementado RSVP-TE sobre el simulador original, como protocolo para la reserva de recursos y para la señalización de LSP. Además, se ha extendido incluyendo las capacidades GLRP de señalización del *GPlane* (mensajes *GReq* y *GPath*) y de recuperación local de paquetes perdidos (mensajes *GReq* y *GAck*), trasladando la señalización GLRP al Plano de Control MPLS. Así, esta información se almacenará en la *GTable*, en lugar de transportar la información de nivel de garantía de servicio mediante apilado de etiquetas, empleando números de etiqueta reservados, o haciendo un tratamiento especial de cada paquete MPLS, como se hace en el simulador original. En el caso de *OpenSimMPLSv2* la *GTable* almacena, para cada flujo prioritario, su *GLevel* y su nodo GLRP anterior en el *GPlane*. El *GBuffer* es el lugar en el que se almacenarán los paquetes pertenecientes a flujos marcados con algún *GLevel*, para lo que *OpenSimMPLSv2* sigue empleando la estructura de memoria DMGP (*Dynamic Memory for GoS PDU*) ya definida en la versión anterior del simulador. Así, si aún queda espacio, se insertarán en el *GBuffer*. En caso contrario, se eliminan paquetes de ese flujo en orden FIFO hasta que se pueda almacenar el nuevo paquete entrante. De este forma, cuanto más tiempo transcurre, más difícil es encontrar en el *GBuffer* los paquetes que se almacenaron hace más tiempo; probablemente cedieron su espacio a paquetes más nuevos. Además, si el nodo ya no posee espacio para nuevos flujos prioritarios GLRP, no se reservará memoria para ellos, por lo que el nodo no podrá retransmitir paquetes de ese flujo. En cierto medida, este nodo todavía podría ofrecer el resto de servicios a ese tráfico que tiene requisitos de GLRP. Por ejemplo, replicar una solicitud de retransmisión a su nodo anterior en el *GPlane*. El tamaño reservado para cada flujo puede ser constante, pero cada nodo estará sometido a un tráfico cruzado diferente. De hecho, el tamaño disponible en el *GBuffer* se asigna por porcentajes del total y siempre según el *GLevel* del flujo, como por ejemplo se muestra en la Tabla 5-1.

Tabla 5-1. Ejemplo de asignación de espacio del *GBuffer* en función del *GLevel*

GLevel	**Espacio asignado**	***GBuffer* = 1KB**	***GBuffer* = 100KB**	***GBuffer* = 1MB**	**...**
1	4%	41B por flujo	4,1KB por flujo	41KB por flujo	...
2	8%	82B por flujo	8,2KB por flujo	82KB por flujo	...
3	12%	123B por flujo	12,3KB por flujo	123KB por flujo	...
...	...	...	...	...	...

Por eso, después de un cierto número de entradas de flujos que hayan reservado espacio, el *GBuffer* no podrá dar servicio a más flujos. Su tamaño depende en gran medida del tráfico que circule por la red, de los requerimientos de los usuarios finales, del tipo de aplicación, etc.

Por otro lado, el funcionamiento del protocolo RSVP-TE implementado en *OpenSimMPLSv2* presenta un inconveniente acerca de la prioridad de los paquetes almacenados en las colas de los puertos. Generalmente, en momentos de congestión, es posible que los mensajes *GReq* recibidos no se procesen inmediatamente, sino cuando un planificador FIFO lo indique. Para evitarlo, *OpenSimMPLSv2* hereda la arquitectura de puertos activos definida en la versión original del simulador, identificando cada tipo de tráfico y asignando prioridades para su tratamiento. En función de esta prioridad, cada paquete irá a una cola distinta (existen 10 colas en el simulador original), de tipo lógico, por lo que pueden tener diferente tamaño, conteniendo cada una tráfico de la misma preferencia. Además, la política de gestión de cada una de estas colas es FIFO, es decir, el tráfico primero se ordena en el puerto por prioridades y posteriormente los paquetes de la misma prioridad se ordenan según el instante de llegada. De esta forma, un algoritmo de tipo *Round Robin* con prioridades accede a las colas en turno circular, leyendo más paquetes de la cola con prioridad 10 y menos paquetes a medida que la prioridad de la cola es menor, repitiéndose el ciclo de forma circular. Así se procesa de forma proporcional más cantidad de paquetes de mayor prioridad que de los menos prioritarios. Hasta cierto punto, esto no implica que sea menos probable descartar los paquetes de mayor prioridad, sino que una vez que el paquete prioritario ha llegado al nodo, será procesado con mayor rapidez, ya que se atienden en mayor proporción los más prioritarios. Realmente, el algoritmo es una variante de SFQ (*Stochastic Fair Queuing*) y de CBQ (*Class Based Queuing*), asignando prioridades en función de la importancia que supondría la pérdida de un paquete de dicho tipo

para el global de la comunicación o para el correcto funcionamiento del dominio MPLS/GLRP.

En particular, *OpenSimMPLS* se ha empleado para investigar acerca del funcionamiento de las redes MPLS [2], [3] y como herramienta para innovar en la docencia universitaria [4], [5], [6]. El estudiante de asignaturas relacionadas con redes y comunicaciones ha visto reforzado su aprendizaje gracias a los ejemplos prácticos incluidos, ya que el simulador ofrece resultados sobre el comportamiento de la red cuando se introduce tráfico prioritario en la red. También permite contrastar resultados gracias al sistema de reconfiguración de los elementos del dominio. De esta forma el alumno puede realizar propuestas para la mejora de supuestos de redes MPLS y detectar los posibles efectos adversos o los beneficiosos sobre el tráfico. Se ha comprobado que el uso del simulador en el aula da lugar a un proceso de doble *feedback*. Por un lado, la interacción con la simulación en ejecución permite al estudiante analizar el comportamiento del dominio de red, obteniendo conclusiones basadas en sus conocimientos teóricos previos y detectando posibles problemas de la fase de diseño del escenario. Por otro lado, tras el análisis de los resultados estadísticos, el alumno también puede obtener conclusiones que redundarán en nuevos cambios de configuración en el dominio MPLS/GLRP. En la Figura 5-2 se muestra este proceso de refinamiento sucesivo, el cual motiva al estudiante a desarrollar sus propias estrategias de pensamiento.

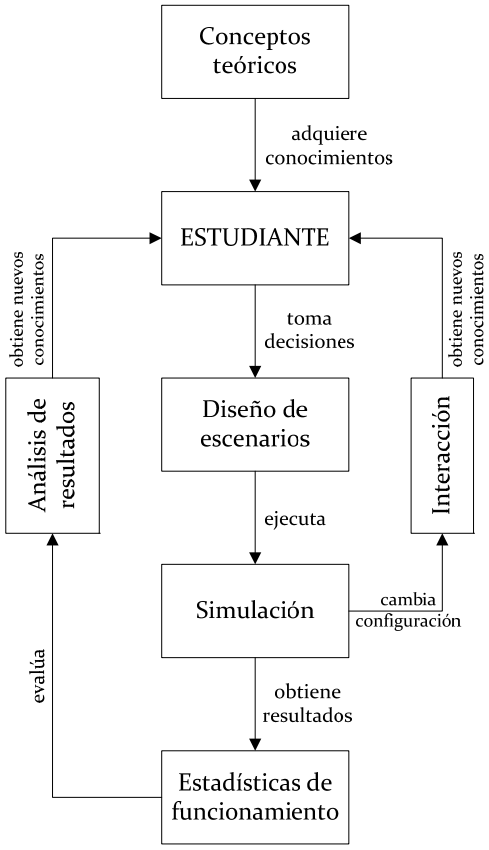

Figura 5-2. Aprendizaje seguido por el estudiante al emplear *OpenSimMPLS*

5.2 Network Simulator

Network Simulator (NS) [7] es un simulador ampliamente aceptado y utilizado en el análisis de redes. Ha sido desarrollado como parte del proyecto VINT (*Virtual InterNetwork Testbed*), proyecto en el que han colaborado la Universidad de California del Sur, el Laboratorio Nacional Lawrence Berkeley, y la Universidad de Berkeley de California. Así, el principal objetivo del proyecto VINT es el de proporcionar una estructura de simulación adecuada para el análisis y el desarrollo de los protocolos de Internet. Permite la simulación de eventos discretos en entornos muy diversos, como protocolos de red y transporte, empleando topologías y agentes, que se definen como puntos de la red donde los paquetes se crean, procesan o reciben. Permite también el uso de diferentes

fuentes de generación de tráfico y simulación de aplicaciones y diversas políticas de gestión de colas y modelos de error.

El código del simulador NS está escrito en C++ y en *Object Tcl* (OTcl). Hay una correspondencia biunívoca entre las clases C++ y las clases OTcl, lo que permite una gran personalización de la simulación de las rutinas (implementadas en C++), así como la configuración y control flexibles de la simulación, empleando un lenguaje sencillo de interpretar, como OTcl. Si bien la flexibilidad de NS proporcionada por dicha dualidad C++/OTcl, se convierte en un aspecto que complica la comprensión de los módulos ya implementados o del desarrollo de otros nuevos, implicando, en la mayoría de casos, trabajar simultáneamente con dos jerarquías de clases relacionadas, dos implementaciones que se entrelazan y dos lenguajes de programación muy diferentes.

Bajo este entorno de desarrollo, ha sido necesaria la implementación de un nuevo módulo para NS2 que ha permitido generar dominios MPLS compatibles con GLRP. Esto da lugar a que, cuando se produzca la perdida de paquetes de flujos prioritarios en un nodo GLRP, éste ya posea toda la información necesaria para iniciar el proceso de solicitud de retransmisión local. Como se ha comentado anteriormente, esto se llevará a cabo desde el Plano de Control MPLS, por lo que se ha modificado NS para que, al mismo tiempo que se configura el LSP, se pueda señalizar el *GPlane*. En particular, se ha debido implementar la *GTable* en los nodos GLRP del dominio, la cual almacenará una fila por cada flujo prioritario que procese, y en la que habrá una referencia al nodo GLRP anterior del *GPlane*. Así, podrá decidir a qué nodo enviar una solicitud de recuperación local en caso de pérdida de paquete o en caso de recibir una solicitud y no haber encontrado el paquete solicitado en su *GBuffer*. Si no hubiera ningún nodo GLRP previo, se trataría entonces del último nodo del *GPlane* y GLRP se detendría. De hecho, el diámetro máximo del *GPlane* se ha implementado mediante la incorporación de un parámetro TTL (*Time to Live*) propio para los mensajes *GReq*.

Para ello, se implementará una extensión de RSVP-TE para configurar las tablas *GTable* de los nodos GLRP del *GPlane* cuando éste se configure, para evitar que la información GLRP deba transmitirse en el Plano de Datos de MPLS, lo que aportaría un exceso de *overhead*. Así, ha sido necesario descartar el empleo del módulo estándar MPLS de NS, ya que éste sólo emplea el protocolo de señalización LDP (*Label Distribution Protocol*) y GLRP requiere RSVP-TE, ya que propone una extensión de éste. En su lugar, se ha optado por MNS (*MPLS Network Simulator*), que es una extensión de NS que añade las

características necesarias para simular dominios MPLS, mejorando el módulo ya existente. Está formado por los componentes CR-LDP, clasificador MPLS, clasificador de servicios, control de admisión, gestor de recursos y planificador de paquetes. Sin embargo, La gran mejora de MNS es el soporte de CR-LSPs (*Constraint based Rounting Label Switching Path*), para tráfico prioritario o con QoS.

Entre otras, MNS posee las siguientes capacidades, adecuadas para el propósito de GLRP:

- Estrategia de activación de LSPs, basada en el Plano de Control o basada en el comportamiento de los flujos de datos ya existentes en la red.
- Esquema para asignación y distribución de etiquetas, que soporta, dentro de los disparadores por control, el esquema *downstream*, y en los disparadores por flujo de datos tanto *downstream* como *upstream*.
- Control de distribución de etiquetas en modo independiente en control, y modos independiente y ordenado en flujo de datos.
- Modo conservativo de retención de etiquetas.
- Rutas explícitas (ER-LSP) con restricciones, basadas en información definida por el usuario.
- Previsión de recursos, en función de los recursos disponibles en los CR-LSPs ya existentes y según la prioridad de cada flujo.
- Agregación de flujos.

Por otra parte, MNS emplea CR-LDP como protocolo de reparto de etiquetas, por lo que se ha empleado una modificación que añade RSVP-TE y las funcionalidades de *DiffServ* sobre MPLS e Ingeniería de Tráfico, desarrollada por IDEO-LABS. Esta nueva implementación ofrece la posibilidad de simular redes MPLS separando los planos de Control y de Datos, es decir, separando los protocolos de señalización MPLS, como RSVP-TE o CR-LDP, de la conmutación de etiquetas o del reenvío de datos, respectivamente. Sobre esta extensión de NS ya fue posible la implementación de GLRP, comenzando por la creación de un nuevo agente de tipo GLRP, que se almacenará en el módulo MPLS ya existente en MNS, para que se pueda obtener dicho agente desde el código OTcl. Además, también ha sido necesario que el *Agente GLRP* almacene el módulo MPLS del nodo donde ha sido asignado.

Tras la creación del *Agente GLRP* en el *Nodo MPLS*, el siguiente paso ha sido hacer llegar al agente el flujo de paquetes de datos que atraviesan el nodo, con el objetivo de hacer una copia de los mismos y almacenarlos en el *GBuffer*, si pertenecen a un flujo prioritario. Posteriormente se reenviarán para que continúen su camino hacia el destino. Para ello, GLRP se inspira en la implementación que hace MNS de RSVP-TE, ya que éste lleva a cabo una captura de los paquetes que contienen alguno de los mensajes de control que define este protocolo, antes de que lleguen al clasificador del *Nodo MPLS*. Así, su objetivo es entregarlos directamente al Agente RSVP-TE situado en dicho nodo, como se muestra en la Figura 5-3. De esta forma, el Agente realizará las acciones oportunas según sea el tipo de mensaje, para entregarlo luego al componente *Nodo MPLS*.

Por tanto, el módulo *RSVPChecker* hará llegar el paquete al *Agente RSVP* situado en el nodo sólo si el paquete contiene un mensaje de tipo RSVP en su campo de datos. Siguiendo esta misma filosofía, se ha implementado la clase *GLRPChecker*, la cual filtrará los paquetes de datos entrantes. Este proceso consiste en comprobar si los paquetes pertenecen a un flujo prioritario, en cuyo caso, si el nodo contiene un *Agente GLRP*, se hará una copia de los mismos para enviarlos al Agente GLRP, como se muestra en la Figura 5-4.

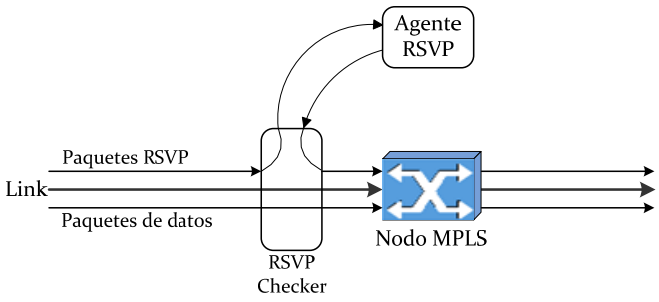

Figura 5-3. Filtrado y captura de paquetes RSVP por parte del *RSVPChecker*.

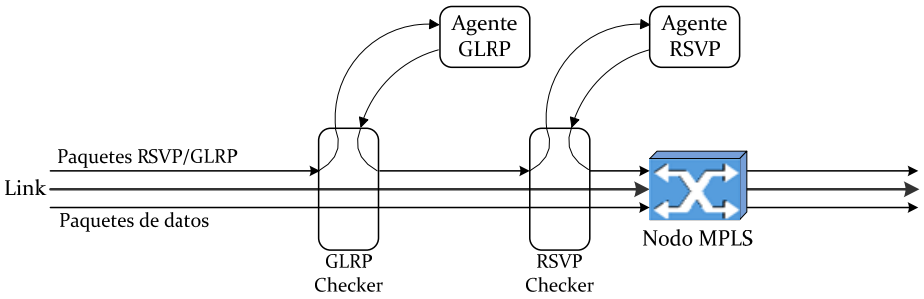

Figura 5-4. Filtrado y captura de paquetes GLRP realizado por el *GLRPChecker*, previo a la acción del *RSVPChecker*

El *Agente GLRP* será el encargado de hacer una copia del paquete y almacenarlo en el *GBuffer*, de forma que si se recibiera un mensaje *GReq* desde un nodo posterior del *GPlane*, se pueda recuperar y reenviar el paquete. De esta manera, tras un intervalo de tiempo determinado, el *GBuffer* contendrá los últimos n paquetes de datos de flujos prioritarios que han pasado por el *Agente GLRP*, siendo n el límite de paquetes que se podrían almacenar. Como ya se ha comentado anteriormente, cuando el *GBuffer* llegue al máximo de su capacidad y reciba un nuevo paquete de un flujo prioritario, se deberá sustituir el paquete de más antigüedad para insertar el nuevo.

El *GBuffer* se ha implementado mediante una lista dinámica de paquetes, instanciando para ello elementos de la clase *Packet*, nativa de NS. Esta lista contará también con el atributo *Tamanio*, el cual definirá el tamaño máximo del *GBuffer*. Además, se ofrece la posibilidad de configurar el tamaño del *GBuffer* del *Agente GLRP* de cada nodo de forma independiente, ya que los nodos GLRP pueden tener necesidades diferentes de almacenar paquetes prioritarios y, por lo tanto, de ofrecer diferentes *GLevel* a los flujos prioritarios que lo atraviesen. Esto puede provocar incluso que durante el control de admisión de RSVP el nodo GLRP ya no tenga capacidad en el *GBuffer* para nuevos flujos y no pueda ofrecer la funcionalidad GLRP. Esto se puede simular configurando su tamaño de *GBuffer* a cero o generando un exceso de tráfico cruzado prioritario a través del nodo. Esto permite simular a los casos en los que el vecino GLRP anterior deja de ser el que satisface las recuperaciones GLRP y pasan a ser otros nodos más remotos del *GPlane* los que encuentran los paquetes solicitados.

Por otro lado, también es necesario que el *Agente GLRP* del nodo que pierde un paquete prioritario detecte las pérdidas, para informar al *Agente GLRP* acerca del enlace en el que se ha producido. Así éste podrá consultar la *GTable* para enviar una solicitud *GReq* al nodo anterior de su *GPlane*. En este sentido se ha empleado la clase *ErrorModel*, nativa de NS para detectar los paquetes que se pierden en función del tráfico cruzado existente en el nodo y de la probabilidad de pérdida asignada desde el fichero de simulación. La clase *ErrorModel* decidirá si cada paquete debe descartarse, siendo en este punto donde se incluirá la extensión GLRP que permite hacer llegar al *Agente GLRP* la información de cada paquete perdido. Así podrá generar un mensaje *GReq* cuyo destino será el *Agente GLRP* del nodo anterior del *GPlane*, como se muestra en la Figura 5-5. Para ello se ha implementado la nueva clase *GLRPMessage*, cuya función es solicitar al *Agente GLRP* que lo reciba, la retransmisión de un paquete perdido, el cual pertenecía a un flujo de datos prioritario. El agente que lo reciba consultará si tiene aún almacenado esos datos para reenviarlos de nuevo al destino. En caso contrario replicará el mensaje de solicitud hacia el *Agente GLRP* de su anterior nodo del *GPlane*, si lo hubiera.

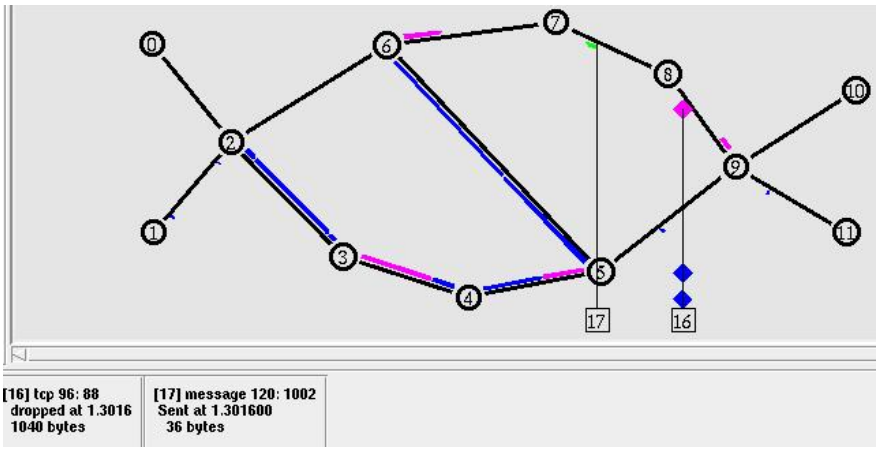

Figura 5-5. Ejemplo de funcionamiento de GLRP sobre NS en el que se aprecia el envío de un mensaje *GReq* (*paquete 17*) desde el nodo 8

5.3 Pruebas realizadas

Como se ha destacado anteriormente, hay varios parámetros que influyen en el rendimiento de GLRP: el tamaño del *GBuffer*, el diámetro de las recuperaciones locales, la velocidad de llegada de nuevos paquetes o la probabilidad de pérdida. Por ello, se llevarán a cabo diferentes simulaciones en las que se modifican todos estos parámetros, con el objetivo de determinar el efecto que tienen sobre métricas como el *delay*, *throughput*, el coste de entrega de paquetes, la ratio de aciertos cuando se solicita un paquete perdido a los nodos del *GPlane* o la eficiencia de GLRP cuando se tiene en cuenta el tráfico cruzado existente en el dominio. En particular, el objetivo es generalizar el comportamiento de GLRP en base a diferentes pruebas, en las cuales se analizan los parámetros involucrados bajo diferentes condiciones o escenarios. Así, se han llevado a cabo una serie de simulaciones para las que se ha caracterizado la red troncal de AT&T (ver Figura 5-6), la cual emplea MPLS para proporcionar QoS a los flujos que demandan servicios de valor añadido. En nuestras simulaciones, el núcleo de la red está caracterizado por 120 nodos LER, 30 nodos LSR y 180 enlaces, con capacidades que varían de 45 Mbps a 2.5 Gbps. En las pruebas llevadas a cabo, la capacidad demandada por cada flujo sigue una distribución dentro del rango de 500Kbps a 2Gbps. Para analizar el efecto que tienen las recuperaciones GLRP sobre la Capa de Transporte, se han comparado el rendimiento de servicios prioritarios que emplean un LSP que incluye algunos nodos GLRP con el de otros flujos no prioritarios a través del mismo LSP, considerando niveles variables de pérdidas entre el 0,01% y el 4%. Dicho LSP consta de 12 saltos, con un RTT de 48ms, el cual es el valor promedio de la latencia de paquetes de los últimos 12 meses medido desde servidores de la red AT&T a otros hosts a nivel global [8]. Los autores de este estudio detallan el método de obtención de dicho valor de RTT, para el que han tomado series de 2.800 mediciones para cada par de ciudades a lo largo de cada mes, obteniendo así la media mensual. En el estudio se defiende esta metodología de medida como una forma más precisa de obtener la experiencia real de usuario, en lugar de emplear herramientas de medición instantánea, como *ping*.

Por otro lado, en todas las simulaciones, cada nodo con capacidad GLRP mantiene un único *GBuffer* para todos los flujos que lo atraviesen, independientemente del número de conexiones establecidas en el nodo en cada instante.

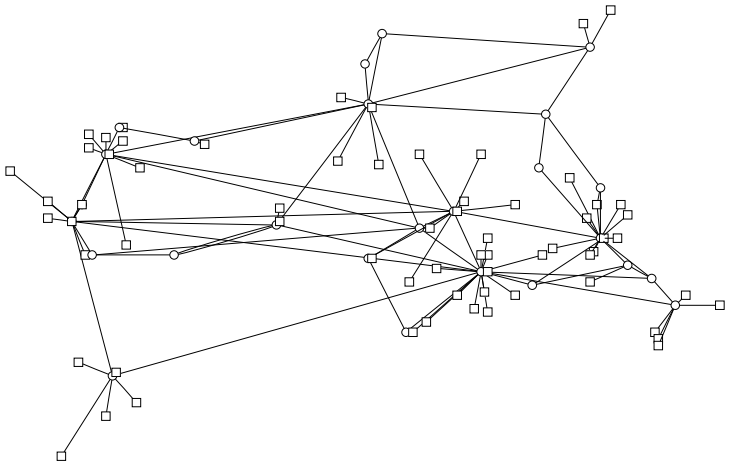

Figura 5-6. Caracterización de la topología de la red AT&T

La mayor parte de los paquetes tienen un tamaño comprendido entre los 40 octetos (el tamaño mínimo de paquete de TCP), los cuales habitualmente transportan mensajes Ack, sin datos, 1500 octetos (el tamaño máximo de trama Ethernet), que es el tamaño típico de las implementaciones de TCP que utilizan mecanismos de descubrimiento de MTU (*Maximum Transmission Unit*), y paquetes de 552 a 576 octetos para las implementaciones de TCP que no emplean descubrimiento de MTU. Así, se ha considerado un tamaño medio de paquete de 425 octetos, con una desviación típica de 521 octetos [9]. De esta forma, en nuestras simulaciones se consideran tres posibles tamaños de paquete: el máximo (1500 octetos), el tamaño promedio (425 octetos) y el mínimo (40 octetos).

Por último, se debe tener también en cuenta que los mensajes de solicitud (*GReq*) y de confirmación (*GAck*), tienen un tamaño de 32 octetos y 36 octetos, respectivamente. Además, se ha considerado un diámetro máximo de *GPlane* de 8 saltos, aunque a lo largo de los diferentes estudios llevados a cabo también se van a configurar varios tamaños de *GBuffer*, para así analizar cómo afecta este parámetro al rendimiento de GLRP.

5.4 Retardo de los paquetes

La Figura 5-7 muestra el retardo de los paquetes en función de la probabilidad de pérdida de paquetes y del diámetro con el que se consiguen las recuperaciones GLRP. El estudio se hace para un flujo particular con una velocidad de 200Mbps. Se puede comprobar que la latencia extremo a extremo de los paquetes aumenta a medida que la probabilidad de pérdida es mayor. Esto se debe a las retransmisiones de los paquetes perdidos. De todos modos, en cualquiera de los casos el retardo de los paquetes recuperados por GLRP es menor que el del caso en el que no se emplea control de errores GLRP, para el que los paquetes perdidos se deben recuperar extremo a extremo desde el nodo origen. En especial y cumpliendo nuestras hipótesis, se aprecia un mejor rendimiento cuando GLRP emplea diámetros pequeños, ya que los paquetes perdidos se recuperan desde una distancia menor. Se puede concluir, por tanto, que la mejora introducida por GLRP en el *delay* de los paquetes de un flujo es directamente proporcional al diámetro de las recuperaciones locales.

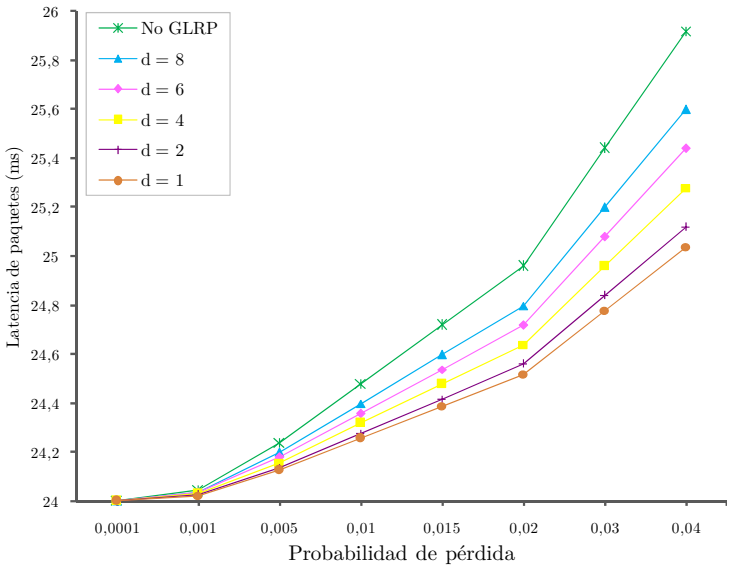

Figura 5-7. Latencia de los paquetes en función de la probabilidad de pérdida para diferentes diámetros de recuperación GLRP

Por otro lado, la Figura 5-8 muestra la latencia de los paquetes para cada posible diámetro de recuperación local y en función de la probabilidad de pérdida. La latencia de nuevo aumenta a medida que se incrementa el diámetro de la recuperación GLRP, ya que la distancia desde donde se recupera cada paquete perdido es mayor. Esto se debe a que, ante la llegada de un mensaje *GReq* a un nodo GLRP, si éste no encuentra el paquete solicitado en su *GBuffer*, replicará esa petición hacia nodos anteriores. Así, con un tamaño de *GBuffer* adecuado, aumentan las posibilidades de recuperar el paquete en un menor número de saltos, dando lugar a un retardo menor. Los resultados también se ven influidos por la probabilidad de pérdida, ya que ésta determina el número de paquetes que se perderán durante el estudio, dando lugar a un mayor número de retransmisiones y, por tanto, a un aumento de la latencia de los paquetes [10]. En cualquier caso, el retardo obtenido al emplear control de errores GLRP sigue siendo menor que la latencia de los paquetes que se deben recuperar extremo a extremo.

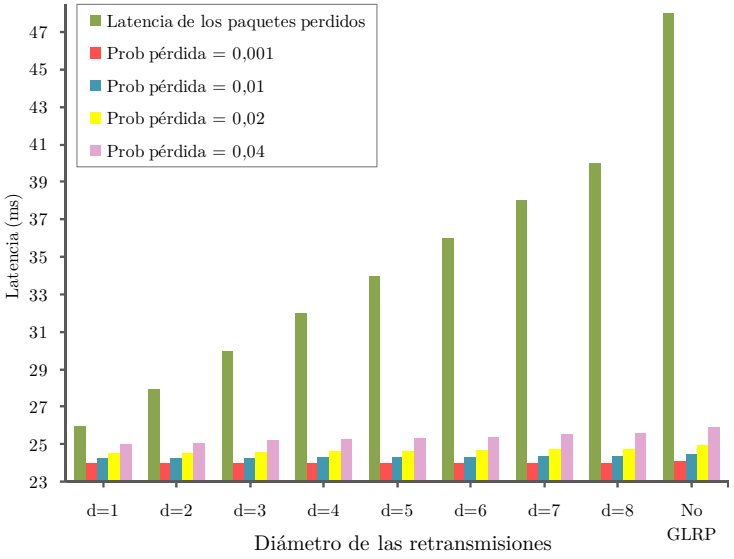

Figura 5-8. Latencia de los paquetes en función del diámetro de recuperación GLRP para diferentes probabilidades de pérdida de paquetes

Por otra parte, también se muestra la latencia de los paquetes perdidos. Aquí se puede apreciar la diferencia de retardo existente al recuperar un paquete empleando un diámetro pequeño de *GPlane* o utilizar una retransmisión extremo a extremo. Este retardo de paquetes perdidos afectará en mayor o menor medida al retardo general de los paquetes en función de la probabilidad de pérdida.

En este caso, si el *GBuffer* empleado es de pequeño tamaño, la probabilidad de necesitar más saltos a lo largo del *GPlane* es mayor, con lo que el retardo para ese paquete se incrementa, ya que RTT_d será mayor. Sin embargo, con protocolos tradicionales en los que no se emplea *GBuffer*, los paquetes descartados sólo pueden retransmitirse desde el extremo origen, lo que implica el recorrido de una ruta más larga para los paquetes retransmitidos, así como la adición de significativos márgenes de tiempo debido a la conservadora naturaleza de los protocolos de capas altas. Como ya sabemos, GLRP permite que estos paquetes descartados puedan encontrarse en el *GBuffer* de algún nodo de su *GPlane* y así puedan recuperarse desde una distancia más corta y retransmitirse hacia el destino. Las gráficas también muestran que los casos GLRP presentan poca variación de retardo al aumentar la probabilidad de pérdida con respecto al caso *No GLRP*, por lo que se presenta también como una interesante opción para servicios prioritarios que requieran una baja variabilidad de retardo.

5.5 Rendimiento de los flujos de datos

La mejora del *delay* de los paquetes introducida por GLRP implica, al mismo tiempo, una mejora del rendimiento o velocidad del flujo de datos. En general, los paquetes emplean menos tiempo para alcanzar el destino, ya que los paquetes perdidos se recuperan desde una distancia más corta. Esto da lugar a que los datos lleguen al receptor más eficientemente, lo que se reflejará en la mejora del *throughput* o del *goodput* del flujo. Esto se aprecia en la Figura 5-9, que presenta un análisis del rendimiento de un flujo a 200Mbps para distintas probabilidades de pérdida de paquetes y diferentes tamaños de *GBuffer*.

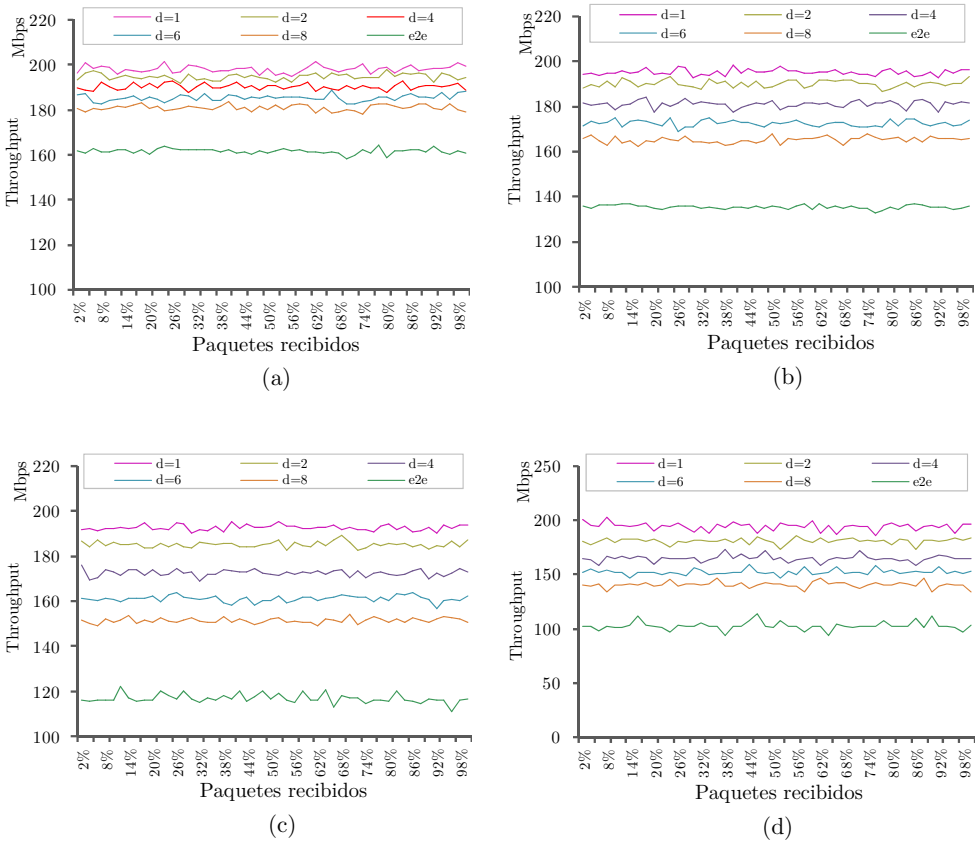

Figura 5-9. Rendimiento de un flujo privilegiado en función del diámetro de recuperación GLRP para diferentes probabilidades de pérdida de paquetes:
(a) *P(loss)*=0,01; (b) *P(loss)*=0,015;
(c) *P(loss)*=0,02; (d) *P(loss)*=0,04

En las gráficas se puede apreciar la diferencia en el *throughput* cuando se emplea un diámetro pequeño en el *GPlane* o cuando se utilizan retransmisiones extremo a extremo para recuperar paquetes perdidos. El tiempo empleado en recuperar los paquetes perdidos afectará en mayor o menor medida al *throughput* general del flujo en función de la probabilidad de pérdida, ya que ésta determina la proporción de paquetes que se perderán durante el estudio. Una probabilidad más alta da lugar a un mayor número de retransmisiones y, por tanto, a un aumento de la latencia de los paquetes, lo que redundará finalmente en un decremento del rendimiento del flujo.

En este caso, al igual que en el estudio del retardo, se observa el mejor *throughput* para los casos de recuperación GLRP que emplean menor diámetro, aunque en todos los casos el rendimiento tiende a ser mejor que el caso No GLRP sin empleo de *GBuffer* y en especial, para los casos de mayor probabilidad de pérdida de paquetes.

5.6 Coste de entrega de paquetes y señalización GLRP

Este estudio muestra el PDC (*Packet Delivery Cost*) junto con el coste de señalización GLRP, con el objetivo de analizar el compromiso existente entre los beneficios aportados por GLRP y el coste que éste supone. El PDC se puede definir como el coste de encaminamiento de los paquetes a través del LSP, siendo un parámetro de especial relevancia para el caso de los paquetes perdidos. En este caso, el PDC refleja el esfuerzo que supone para la red la retransmisión de dichos paquetes [11]. A este parámetro se le une el coste de señalización GLRP, de manera que por un lado GLRP introduce en la red paquetes de señalización, pero por otro permite reducir el PDC.

En la Figura 5-10 se muestra la unión de ambos parámetros, en función de la probabilidad de pérdida y del diámetro de recuperación GLRP para un flujo a 200Mbps. Se puede comprobar, por ejemplo, que para tamaños de paquete de 1500 ó 425 octetos GLRP sigue ofreciendo un menor coste de reenvío de paquetes para la red. Sólo en el caso de paquetes de 40 octetos, que es el tamaño mínimo de paquete, GLRP deja de aportar mejora a partir de diámetros superiores a 4 saltos. Esto se debe a que la cantidad de señalización introducida por GLRP en la red se vuelve significativa a partir de esa distancia en proporción a los paquetes de datos, que son muy pequeños. Se puede considerar, por tanto, que para mantener el equilibrio entre los beneficios de GLRP y el posible *overhead* que introduce en la red, no se debería configurar un *GPlane* con diámetro superior a 4 saltos si todos los paquetes van a ser de tamaño mínimo.

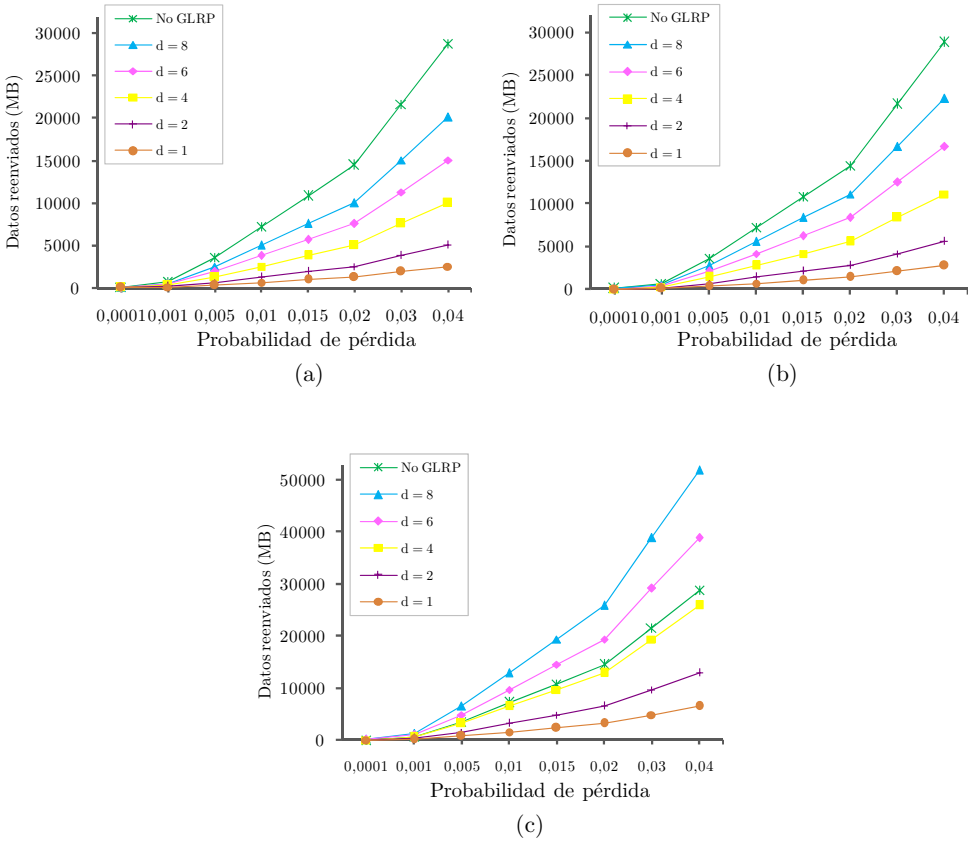

Figura 5-10. PDC + señalización GLRP en función del tamaño de paquete: (a) *Tamaño* = 1500 octetos; (b) *Tamaño* = 425 octetos; (c) *Tamaño* = 40 octetos

La Figura 5-11 muestra los resultados del mismo estudio sin tener en cuenta el tamaño de paquete, sino que sólo se considera el número de paquetes transmitidos. El estudio se hace, también, en función de la probabilidad de pérdida y del diámetro de recuperación GLRP. De nuevo, a partir de diámetros superiores a 4 saltos GLRP deja de aportar mejora con respecto a las retransmisiones extremo a extremo (caso *No GLRP*).

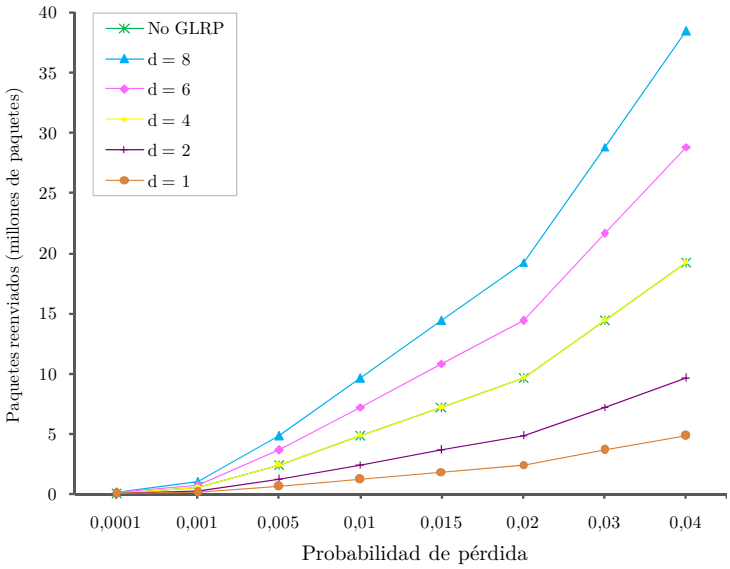

Figura 5-11. PDC + señalización GLRP según paquetes transmitidos.

En este punto, el número de paquetes de señalización, junto con el número de saltos de los paquetes retransmitidos por GLRP, se iguala con el número de reenvíos del caso *No GLRP*, aunque como se pudo comprobar en el estudio anterior, a pesar de igualarse en número de paquetes, la cantidad de datos que reenvía GLRP sigue siendo menor. Además, es de destacar que GLRP ha hecho uso de los mensajes de respuesta *GAck*, aunque su empleo es opcional.

5.7 Ratio de aciertos del *GBuffer*

En este estudio se analiza la efectividad del *GBuffer*, calculada como el número de veces que se encuentra un paquete solicitado al acceder al *GBuffer* de un nodo dividido por el número de veces que se accede a dicho *GBuffer*. Esta métrica representa el índice de aciertos de *GBuffer* en nodos situados a diferentes diámetros, para distintos tamaños de *GBuffer* y diferentes velocidades. El objetivo es analizar la relación existente entre la probabilidad de encontrar un paquete perdido, la velocidad de llegada de nuevos paquetes de datos, el diámetro de las recuperaciones GLRP y el tamaño de *GBuffer*. Esto permitirá obtener el tamaño ideal de *GBuffer* para cada velocidad y diámetro máximo admisible.

A priori, para optimizar el funcionamiento de GLRP lo idóneo sería configurar un *GBuffer* de gran tamaño en los nodos, ya que así más paquetes podrían almacenarse en un instante dado. No obstante, independientemente de las elevadas capacidades de almacenamiento de los routers comerciales actuales, un nodo GLRP no requiere almacenar un elevado número de paquetes, ya que los nuevos paquetes entrantes pueden sobrescribir a los más antiguos, cuando estos ya no tengan posibilidad de ser solicitados.

Así, la optimización del tamaño del *GBuffer* tiene, como primer objetivo, conseguir búsquedas de paquetes más rápidas. En este sentido, el tamaño óptimo de buffer de un nodo GLRP para un flujo particular se puede calcular como *Ratio_paquetes* x RTT_d. Este es el producto de la velocidad de llegada de nuevos paquetes por el RTT_d, que es el *round trip time* para un diámetro d y que, como ya se ha descrito, es el tiempo transcurrido desde que se reenvía un paquete desde el nodo GLRP que almacena un paquete, hasta un nodo d saltos más adelante y que descarta el paquete. A esto se le añade el tiempo que transcurre hasta que llega la solicitud *GReq* al nodo GLRP que almacenó el paquete, desde el nodo que lo descartó. En este sentido, si el tamaño de *GBuffer* para un flujo es sensiblemente menor que *Ratio_paquetes* x RTT_d, la probabilidad de encontrar el paquete solicitado en los nodos del *GPlane* se reduce [12].

Esta hipótesis se ve apoyada por los resultados de las simulaciones, como se aprecia en la Figura 5-12. En ella se muestra el índice de aciertos a la hora de localizar un paquete en el *GBuffer* de alguno de los nodos GLRP, teniendo en cuenta un diámetro máximo de 8 saltos. De esta forma, la ratio de aciertos experimenta una mejora significativa a medida que el tamaño del *GBuffer* se acerca al óptimo. Es más probable encontrar un paquete para su retransmisión GLRP en un nodo cercano si el tamaño de *GBuffer* es el adecuado, evitando así el reenvío de mensajes *GReq* hacia nodos anteriores del *GPlane*.

Por consiguiente, existe una cierta localidad temporal en el *GBuffer* de los nodos del *GPlane*. Cuando un nodo es incapaz de reenviar el tráfico hacia el siguiente salto debido a la congestión existente, se producirá la pérdida de uno o varios paquetes. En cierto modo, si se trata de un nodo GLRP, podrá enviar una solicitud de retransmisión local a su nodo previo en el *GPlane*. Cuando éste reciba el mensaje accederá a su tabla *GIndex* para comprobar si aún tiene almacenado el paquete solicitado, para recuperarlo y proceder a su reenvío hacia el destino.

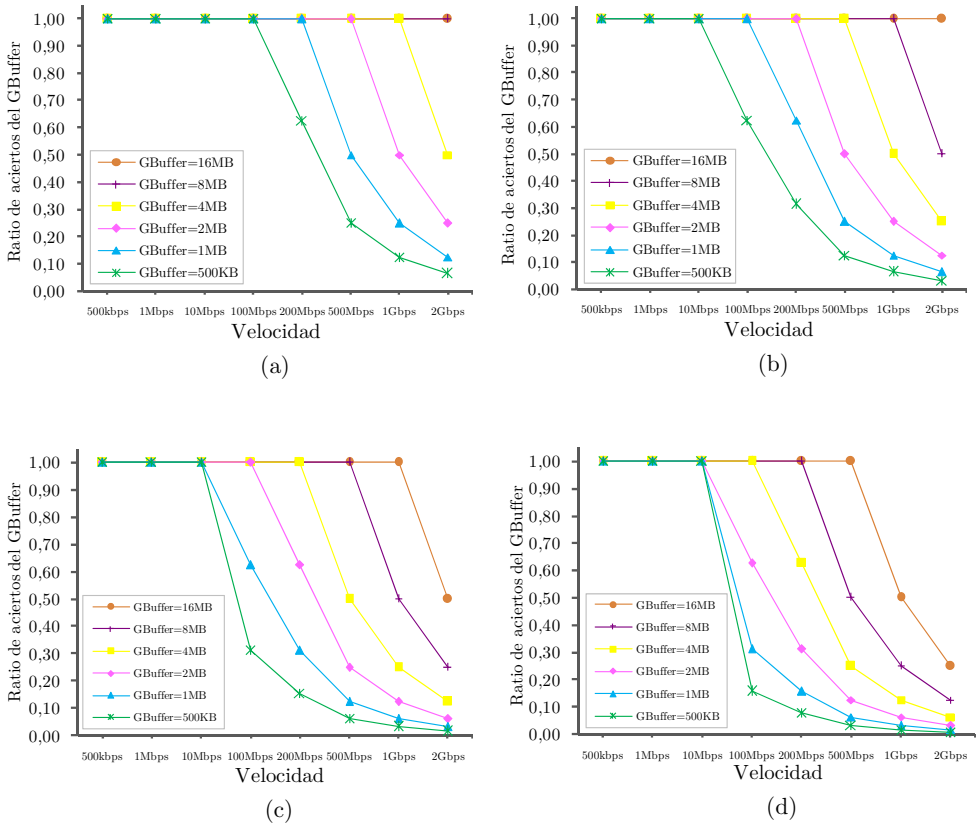

Figura 5-12. Índice de aciertos en función de la velocidad y tamaño del *GBuffer*, para varios diámetros de *GPlane:*
(a) $d = 1$; (b) $d = 2$; (c) $d = 4$; (d) $d = 8$

De ahí que, gracias a esta propiedad de localidad temporal, no es necesario un tamaño de *GBuffer* elevado, ya que sólo los paquetes almacenados recientemente serán los solicitados y utilizados en la retransmisiones GLRP. Además, un *GBuffer* excesivamente grande tampoco aporta mejora, ya que los paquetes almacenados por un tiempo superior a RTT_d nunca serán solicitados, por lo que habrán sido sobrescritos por los nuevos paquetes entrantes.

Por otro lado, el tamaño también debe analizarse teniendo en cuenta el diámetro de las recuperaciones locales, así como el nivel de congestión existente en el dominio. Si se analiza en las gráficas la ratio de aciertos en función del diámetro de recuperación GLRP descubrimos que, si se pudieran garantizar recuperaciones GLRP con diámetros pequeños, la ratio sería alta a pesar de que

el tamaño del *GBuffer* sea inferior al óptimo. Es decir, si los paquetes perdidos se pudieran recuperar empleando diámetros pequeños, se podrían incluso emplear tamaños de *GBuffer* inferiores al óptimo, ya que el parámetro RTT_d es directamente proporcional al diámetro. Por este motivo las figuras muestran una ratio de aciertos muy elevada cuando el diámetro es de sólo 1 ó 2 saltos, a pesar de disponer de poco espacio en el *GBuffer*. Esto se debe a que transcurre poco tiempo desde que se almacena el paquete hasta que se recibe el mensaje *GReq*. En este caso los nuevos paquetes entrantes no han sobrescrito todavía aquellos que van a ser solicitados. En cierto modo, si el diámetro necesario para conseguir la retransmisión GLRP es mayor, la probabilidad de que los nodos a los que se solicita la retransmisión local tengan todavía almacenados los paquetes descartados es menor, ya que RTT_d es mayor. Así, para tamaños de *GBuffer* inadecuados, si los paquetes descartados no se encuentran en los nodos más cercanos, será poco probable que se localicen en nodos anteriores más lejanos, ya que al transcurrir más tiempo la probabilidad de que hayan sido sobrescritos es mayor.

5.8 Recuperaciones GLRP en función del tráfico cruzado

Este estudio muestra el porcentaje de paquetes recuperados desde nodos situados a diferentes diámetros, con respecto al total de paquetes perdidos. En este caso se ha tenido en cuenta también el tráfico cruzado, que es el que hace que el *GBuffer* de algunos nodos esté más saturado que el de otros, provocando incluso que se recuperen más paquetes desde nodos más distantes del *GPlane*. La extensión desarrollada sobre MNS ha permitido integrar GLRP en un dominio MPLS, para generar el tráfico cruzado que atraviesa cada nodo. A partir de aquí, la Figura 5-13 hace una comparativa entre los porcentajes de paquetes recuperados desde cada diámetro. También muestra el porcentaje de paquetes que no ha podido recuperar GLRP, para diferentes velocidades. Así, se puede apreciar, por ejemplo, que el nodo situado a un diámetro de un salto soporta un mayor tráfico cruzado prioritario. Esto ha hecho que su *GBuffer* esté más saturado que los nodo situados a dos o cuatro saltos, lo que da lugar a que se recuperen más paquetes desde esos nodos que desde el vecino GLRP anterior.

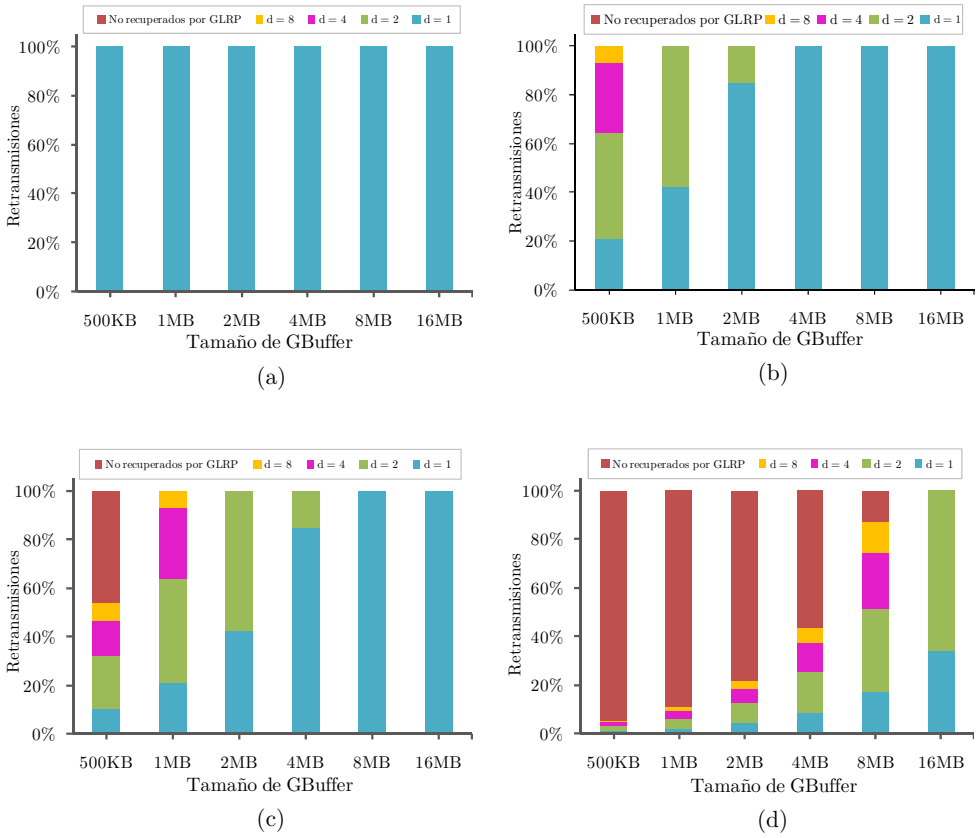

Figura 5-13. Retransmisiones desde cada nodo en función del tamaño de *GBuffer* y de la velocidad:
(a) Velocidad = 10Mbps; (b) Velocidad = 100Mbps;
(c) Velocidad = 200Mbps; (d) Velocidad = 2Gbps

El estudio refleja también la relación existente entre el tamaño idóneo de *GBuffer* y la velocidad de llegada de paquetes. Por ejemplo, en el caso (a), para un flujo a 10Mbps, un *GBuffer* con tamaño de 500KB, el cual es un tamaño muy por debajo del ideal ya analizado (*Ratio_paquetes* x RTT_d), es suficiente para garantizar una recuperación GLRP óptima de los paquetes perdidos; es decir, desde el nodo anterior más cercano del *GPlane* (d=1). En cierto modo, a medida que la velocidad del flujo es mayor, GLRP necesita hacer solicitudes a otros nodos más lejanos en el *GPlane*. En el caso (b), para un tamaño de 500KB, sólo es posible recuperar alrededor del 20% de los paquetes perdidos desde el nodo situado a un diámetro de un salto, pero el resto de paquetes todavía puede ser

recuperado por GLRP. Hasta cierto punto, en los casos (c) y (d) existe un porcentaje importante de paquetes que no van a poder ser recuperados por GLRP si se emplea un tamaño de 500KB. En estos casos la retransmisión quedaría en manos de la Capa de Transporte. Esto se puede evitar, como se observa en la figura, al emplear tamaños de *GBuffer* más adecuados, que se acerquen al tamaño ideal hasta, por ejemplo, el caso (d) de un flujo a una velocidad de 2Gbps, en el que un tamaño de 8MB permite recuperar localmente casi el 90% de los paquetes perdidos, o con 16MB se recupera el 100% desde nodos situados a diámetros de uno o dos saltos.

5.9 Referencias del capítulo

[1] F. J. Rodríguez-Pérez and J. L. González-Sánchez, "Guarantee of Service (GoS) support over MPLS using Active Techniques," WSEAS Transactions on Computers, vol 3, nº 6. Pag 1959-1964. [Online]: http://dl.acm.org/citation.cfm?id=1374221 (último acceso julio 2014).

[2] M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. L. González-Sánchez, J. L. Marzo and A. Gazo, "An Architecture to provide Guarantee of Service (GoS) to MPLS," Proceedings of the IV Workshop in MPLS/GMPLS networks, Pag. 139-149. 2005.

[3] M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. L. González-Sánchez and A. Gazo, "Multiplatform and Opensource GoS/MPLS Simulator, "Proceedings of the II European Modeling and Simulation Symposium (EMSS2006). International Mediterranean Modelling Multiconference (I3M2006). Pag. 529-537. 2006.

[4] F. J. Rodríguez-Pérez, M. Domínguez-Dorado, J. L. González-Sánchez, J. L. Marzo Lázaro and A. Gazo-Cervero, "*OpenSimMPLS*: Herramienta para la Innovación Docente e Investigación en Redes y Comunicaciones," V Jornadas de Ingeniería Telemática (JITEL'05). Pag. 87-94. 2005.

[5] M. Domínguez-Dorado, F. J. Rodríguez-Pérez and J. L. González-Sánchez, "Simulador MPLS para la Innovación Pedagógica en el Área de Ingeniería Telemática," IEEE RITA. Revista Iberoamericana de Tecnologías del Aprendizaje, vol 2, nº 1. Pag 27-34. 2007.

[6] M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. Carmona-Murillo and J. L. González-Sánchez, "Educational improvements applying an MPLS network simulator: a technical approach," IEEE Multidisciplinary Engineering Education Magazine, vol 2, nº 4. Pag 18-25. 2007.

[7] T. Issariyakul and E. Hossain, "Introduction to Network Simulator NS2," Ed. Springer-Verlag. 2009. ISBN 978-0-387-71759-3.

[8] L. Ciavattone, A. Morton and G. Ramachandran, "Standardized Active Measurements on a Tier 1 IP Backbone," IEEE Comm. Magazine, vol 41, nº 6. Pag 90-97. 2003. [Online]: http://dx.doi.org/10.1109/MCOM.2003.1204753 (último acceso julio 2014).

[9] Fang Liu, Xiaolong Wu, Weimin Li and Xiaonan Liu, "The packet size distribution patterns of the typical Internet applications," IEEE International Conference on Network Infrastructure and Digital Content. Pag. 325-332. 2012. [Online]: http://dx.doi.org/10.1109/ICNIDC.2012.6418769 (último acceso julio 2014).

[10] J. Carmona-Murillo, J.L. González-Sánchez, D. Cortés-Polo and F.J. Rodríguez-Pérez, "DM3: distributed mobility management in MPLS-based access networks," International Journal of Network Management, vol 24, nº 2. Pág. 85-100. 2014. [Online]: http://dx.doi.org/10.1002/nem.1854 (último acceso abril 2014).

[11] D. Cortés-Polo, J.L. González-Sánchez, F.J. Rodríguez-Pérez and J. Carmona-Murillo, "Mobility management in packet transport networks for network convergence," Transactions on Emerging Telecommunications Technologies, vol [pendiente], nº [pendiente]. Pag [pendiente]. 2013. [Online]: http://dx.doi.org/10.1002/ett.2705 (último acceso abril 2014).

[12] F.J. Rodríguez-Pérez, J.L. González-Sánchez, J. Carmona-Murillo and D. Cortés-Polo, "An OAM function to improve the packet loss in MPLS-TP domains for prioritized QoS-aware services," International Journal of Communication Systems, vol [pendiente], nº [pendiente]. Pag [pendiente]. 2014. [Online]: http://dx.doi.org/ 10.1002/dac.2742 (último acceso abril 2014).

Capítulo 6. Conclusiones y líneas futuras

La ciencia, a pesar de sus progresos increíbles, no puede ni podrá nunca explicarlo todo. Cada vez ganará nuevas zonas a lo que hoy parece inexplicable. Pero las rayas fronterizas del saber, por muy lejos que se eleven, tendrán siempre delante un infinito mundo de misterio.
Gregorio Marañón

En este capítulo se resumen las principales conclusiones que se desprenden de la presente Tesis Doctoral, así como diversas líneas de trabajo futuras.

En la presente Tesis Doctoral se ha hecho una propuesta, denominada GLRP, para la mejora del rendimiento de flujos prioritarios en dominios MPLS-TP sujetos a congestión. Para ello, se analizaron previamente los esquemas de control de congestión más importantes propuestos hasta la fecha. Entre ellos se han destacado mecanismos de control adaptativo del ancho de banda, de predicción del tráfico, de gestión activa de colas o de control de la congestión basándose en el *feedback* de la propia red.

Por un lado, se ha propuesto una solución para mejorar el proceso de detección de pérdidas, el cual no está basado en el *feedback* proporcionado por la red mediantes confirmaciones Ack desordenadas, sino mediante notificaciones explícitas entre nodos cercanos.

Por otro lado, una segunda solución mejora el tiempo empleado en recuperar los datos perdidos y hacerlos llegar a su destino, al emplear una ruta más corta para ello. Se ha presentado como una solución escalable y distribuida que propone, para su integración con MPLS-TP, una extensión del protocolo de

señalización RSVP-TE. En particular, se ha desarrollado una extensión al proceso de señalización del LSP que lleva a cabo RSVP-TE. Esto permite que el LSP pueda incluir uno o varios conjuntos de nodos cooperantes para el control local de la pérdida de paquetes. Estos conjuntos de LSR se han denominado *GPlane* y su creación se lleva a cabo durante la creación del LSP que los contiene. Así, esta cooperación que se establece entre los nodos intermedios de un *GPlane* permitirá la recuperación local del tráfico perdido de los flujos más prioritarios. Para ello, se almacenarán temporalmente los paquetes de flujos particulares en dichos nodos. Así, se habilitan más nodos desde los que recuperar paquetes perdidos, de forma que la única configuración necesaria consiste, básicamente, en que cada nodo con capacidad GLRP del *GPlane* conozca la dirección del nodo GLRP anterior, al cual enviará una solicitud de retransmisión local cuando detecte la pérdida de un paquete perteneciente a algún flujo prioritario.

Se han analizado también algunas funciones adicionales de GLRP, relacionadas con servicios no orientados a conexión, con la pérdida de ráfagas de paquetes, con la reordenación de los paquetes retransmitidos o con el comportamiento de GLRP en redes de distribución punto-multipunto.

Se ha demostrado analíticamente que los tiempos de detección de pérdidas, así como el tiempo empleado en recuperar dichos datos es sustancialmente menor al emplear GLRP, comparado con las retransmisiones extremo a extremo propias de mecanismos de Capa de Transporte. Así, se ha analizado comparativamente el consumo de recursos, el retardo de los paquetes, ancho de banda consumido o la memoria necesaria. Estos parámetros están involucrados en la notificación de las pérdidas, en el envío de las solicitudes de retransmisión local o en el almacenamiento de paquetes de flujos prioritarios. Además, se ha presentado, por un lado, la implementación de *OpenSimMPLS* v2, siendo apto para fines de innovación docente y de investigación y, por otro, una extensión del conocido simulador *Network Simulator*, con el fin de obtener resultados de simulación del funcionamiento de GLRP.

Como conclusión, esta Tesis Doctoral propone un esquema distribuido de gestión de pérdidas de datos prioritarios en capas bajas y como apoyo al mecanismo convencional de Capa de Transporte. Abarca diferentes posibilidades de funcionamiento, teniendo en cuenta servicios no orientados a conexión, ráfagas de pérdidas de paquetes, la posible distribución de datos en rutas punto-multipunto y la necesaria reordenación de los paquetes retransmitidos. Así mismo, los resultados obtenidos han sido publicados como artículos en diversos

congresos y revistas de ámbito internacional, entre los que se destacan los siguientes:

- Francisco Javier Rodríguez Pérez; José Luis González Sánchez; Javier Carmona Murillo; David Miguel Cortés Polo, "An OAM function to improve the packet loss in MPLS-TP domains for prioritized QoS-aware services", International Journal of Communication Systems (ISSN: 1074-5351). Enero 2014. DOI: 10.1002/dac.2742. JCR (2013): 1.106.
- Francisco Javier Rodríguez Pérez; José Luis González Sánchez; David Miguel Cortés Polo; Javier Carmona Murillo, "A Delay-Oriented Prioritization Policy Based on Cooperative Lossless Buffering in PTN Domains", Journal of Network and Systems Management (ISSN: 1573-7705). Octubre 2014. DOI: 10.1007/s10922-014-9334-4. JCR (2013): 0.438.
- Francisco Javier Rodríguez Pérez; José Luis González Sánchez; Alfonso Gazo-Cervero, "RSVP-TE Extensions to Provide Guarantee of Service to MPLS", 6th IFIP Networking Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet. Vol. 4479 of Lecture Notes in Computer Science. Pág. 808-819. Springer-Verlag. Mayo 2007. DOI: 10.1007/978-3-540-72606-7_69.
- David Miguel Cortés Polo; José Luis González Sánchez; Francisco Javier Rodríguez Pérez; Javier Carmona Murillo, "Mobility management in packet transport networks for network convergence", Transactions on Emerging Telecommunications Technologies) (ISSN: 1124-318X). Septiembre 2013. DOI: 10.1002/ett.2705. JCR (2013): 1.354.
- Javier Carmona Murillo; José Luis González Sánchez; David Miguel Cortés Polo; Francisco Javier Rodríguez Pérez, "DM3: distributed mobility management in MPLS-based access networks", International Journal of Network Management (ISSN: 1099-1190), 85-100. Diciembre 2013. DOI: 10.1002/nem.1854. JCR (2013): 0.517.

Además, la continuación del trabajo desarrollado en esta Tesis Doctoral está dirigida a publicar resultados adicionales en otras revistas, así como enviar algún *Internet Draft* que explique la metodología y los resultados del trabajo, como por ejemplo la propuesta de extensión de RSVP-TE para permitir la creación del *GPlane* en el LSP. Junto a la publicación de los resultados continuarán los trabajos de investigación, con el objetivo de ampliar el marco de los escenarios en los que se pueden aplicar los mecanismos aquí propuestos, empleando, en la medida de lo posible, redes en explotación, o bien las matrices de rendimiento

obtenidas de las mismas. Para ello, se desarrollará un modelo que permita emplear informes de tráfico o estadísticas a los que aplicar GLRP. Los resultados obtenidos se podrán luego generalizar y extrapolar a otros dominios o al rendimiento futuro del mismo dominio, en función de la evolución que sigan los parámetros de tráfico estudiados de la red en cuestión.

Por otro lado y analizando las tendencias más recientes, se puede comprobar que existe un gran incremento del tráfico global en redes de comunicación móvil. Así mismo, algunos servicios como la Televisión IP (IP-Tv) y el vídeo bajo demanda (VoD) se desplegarán como servicios masivos para todos los usuarios de redes móviles, lo que tendrá un impacto aún mayor sobre el tráfico de red. Para adaptar la red a este incremento de tráfico se deben resolver tres importantes puntos: la integración de las redes heterogéneas, el mantenimiento de la conectividad con la red y la gestión de los recursos requeridos por el nodo móvil. En este contexto, GLRP se integrará con otras propuestas, como técnica de *buffering*, con el objetivo de minimizar la pérdida de paquetes durante el proceso de *handover*, siendo así una alternativa más eficiente que, por ejemplo, el uso de túneles para retransmitir los paquetes perdidos o la transmisión simultánea a las dos estaciones base implicadas en el *handover*.