Universidad de Alcalá

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Automática

Contribución al Diseño de Conmutadores Transparentes Avanzados Basados en Tecnología Ethernet

TESIS DOCTORAL

Juan Antonio Carral Pelayo

Alcalá de Henares, abril 2013

Juan Antonio Carral Pelayo
Área de Ingeniería Telemática (Departamento de Automática)
Escuela Politécnica Superior de la Universidad de Alcalá
Campus Universitario. Ctra. Madrid-Barcelona, Km. 31,600
Alcalá de Henares (Madrid)
e-mail: juanantonio.carral@uah.es

Dedicatoria

A mi familia, ellos están ahí siempre.

Este trabajo ha sido parcialmente financiado por la Junta de Comunidades de Castilla la Mancha (JCCM) y por la Comunidad de Madrid a través de los proyectos EMARECE (PII1I09-0204-4319) y MEDIANET (S2009/TIC-1468).

Agradecimientos

Deseo expresar mi agradecimiento a todos los que han hecho posible la elaboración de esta Tesis.

En primer lugar a mí tutor sin cuya fe y obstinación, más allá del desaliento, no hubiera sido posible.

A los demás compañeros del grupo de investigación GIST-Netserv y de la Unidad Docente de Ingeniería Telemática, del pasado y del presente, por su colaboración y disposición, cada uno en la medida de sus posibilidades.

Por último, esto ha sido una carrera de fondo que comenzó hace mucho tiempo, y no quiero olvidar al grupo de redes satélite del Departamento de Ingeniería de Servicios Telemáticos de la UPM. Ellos me formaron como investigador y como profesor, a ellos les debo mucho de lo que sé y de lo que soy.

Resumen

Aunque los conmutadores Ethernet son ya el elemento clave en las redes actuales campus, empresariales y de centros de proceso de datos por sus altas prestaciones, coste moderado y mínima configuración, los protocolos de capa dos actuales no tienen la escalabilidad y robustez suficientes para utilizarse en redes campus de tamaño medio y precisan el uso de encaminadores (*routers*) que compartimenten la red. Estas redes de tamaño y capacidad crecientes requieren nuevos dispositivos que superen las limitaciones de escalabilidad de los puentes y la complejidad de configuración de los encaminadores. Desde 2004 se vienen estandarizando dos propuestas divergentes aunque ambas basadas en introducir encaminamiento por estado de enlaces en capa dos: TRILL y Shortest Path Bridges. Asimismo han aparecido diversas propuestas propietarias de fabricantes e investigadores. Pero aún no existe consenso sobre la adecuación de dichas soluciones a los problemas planteados por la escalabilidad de Ethernet en los escenarios mencionados debido a las dificultades del problema planteado de hacer Ethernet escalable manteniendo la compatibilidad y la simplicidad en la arquitectura.

Este trabajo aporta varias contribuciones en el campo de los conmutadores Ethernet Avanzados para redes campus y centros de datos que se enmarcan en una línea propia de investigación de conmutadores Ethernet auto-configurables, desarrollada en la Universidad de Alcalá en los últimos seis años, que, partiendo de la investigación en protocolos de prohibición de giros y de encaminamiento jerárquico ha desembocado en la reciente arquitectura de conmutadores All-Path/ARP-Path que recupera la simplicidad de los puentes a través de su evolución como tales en vez de la hibridación con protocolos de encaminamiento. Entre las contribuciones de la Tesis se incluyen protocolos de Ethernet de encaminamiento basados en árbol y protocolos jerárquicos *Up/Down* y múltiples contribuciones a la definición, especificación, análisis y validación de la nueva familia de protocolos All-Path.

Palabras clave: redes de computadores, protocolos, puentes transparentes, árboles múltiples de expansión, prohibición de giros, direccionamiento jerárquico.

Abstract

Current features of Ethernet switches like high performance, low cost and zero configuration make them suitable for big campus networks, but standard layer two protocols do not scale to medium size campus networks without routers to segment the switching domain. These networks, with its increasing size, bandwidth and complexity, require new devices that overcome the limitations of current switches and the complexity of configuration of routers. Two different approaches, based on link-state routing paradigm, have been under investigation at IETF and IEEE since 2004, leading to the TRILL Routing bridges and IEEE Shortest Path Bridges solutions. In the mean time, many other proposals coming both from industry (thus proprietary) and from research bodies have been studied. However, there is not yet a clear consensus about the suitability of all this proposals to solve the scalability issue while maintaining, at the same time, backwards compatibility and simplicity.

This work presents several contributions on the field of *Advanced Ethernet Switches,* designed for campus and data-center networks. It is an integral part of the research line on self-configuring Ethernet switches, developed at the University of Alcalá since 2006. It all starts from research activities on turn-prohibition routing protocols and hierarchical routing and has, just recently, produced a new switching architecture called All-Path (Arp-Path). All-Path can be thought of as an evolution of the classical transparent bridging paradigm rather than as an hybridization of Ethernet switching and a routing protocol. The main contributions of this PhD. Thesis include new spanning tree based, routing protocols, a hierarchical up/down (turn prohibition) routing protocol and many other contributions to the definition, specification and characterization of the family of All-Path protocols.

Keywords: transparent bridges, protocols, spanning tree, routing, shortest path bridges

Índice general

Índice de figuras

Índice de tablas

1 Introducción y objetivos

En este capítulo se presenta el marco de la tesis, sus objetivos y estructura. La demostrada versatilidad y capacidad de adaptación de Ethernet a distintos entornos de aplicación con requisitos cambiantes y, en muchos casos, hasta contradictorios, ha catapultado la tecnología hasta convertirla en dominante en entornos impensables hace muy pocos años. No sólo el constante aumento de prestaciones en términos de velocidad de transmisión a un coste razonable, sino también la capacidad para escalar desde el entorno clásico, y reducido, de una red local, para abarcar cada vez redes más y más grandes, donde no solo el factor tamaño es importante, también la capacidad para gestionar distintos tipos de tráfico, con sus correspondientes requisitos, como es el caso de las redes metropolitanas y de proveedor. Cabe destacar también la creciente importancia de los centros de proceso de datos donde una vez más Ethernet ha encontrado un entorno de aplicación prometedor, donde la capacidad de conmutación a altas velocidades de ingentes cantidades de tráfico con equipos de reducido coste, configuración y gestión van a resultar determinantes. Se resume la evolución reciente de las redes basadas en tecnología Ethernet. Se identifican los principales aspectos que, actualmente, limitan su aplicación en nuevos entornos de servicio, para plantear el problema que se pretende resolver con el trabajo realizado en la Tesis y se adelantan las líneas principales que han guiado el trabajo. Finalmente se describe la organización de la Tesis.

1.1 Introducción

Ethernet ha pasado en apenas 30 años, gracias a sus altas prestaciones (hasta 40 Gbps actualmente y 100 Gbps en estandarización), su gran variedad de velocidades (desde 100 Mbps a 10 Gbps) en interfaces estándar de bajo coste que le permite adaptarse a distintos entornos y necesidades y, sobre todo, sus características de autoconfiguración y compatibilidad con instalaciones previas, de ser una tecnología LAN a ser la tecnología de interconexión y de transporte dominante, sustituyendo sucesivamente a otras tecnologías más prometedoras o, incluso, ya asentadas en ciertos entornos como FDDI, ATM y SDH. Esta evolución parece no tener fin y amenaza con convertirla en la tecnología dominante en el mundo de las redes de datos independientemente del entorno de aplicación, del tamaño de la red o del servicio que se necesite proveer a los clientes.

1.2 La tecnología Ethernet

Ethernet se ha consolidado en los últimos años como el estándar de interconexión en las redes locales de ámbito empresarial y metropolitano. Ha desplazado a otras tecnologías como FDDI, ATM, SDH y otros, por sus altas prestaciones, compatibilidad con equipos previos (*legacy)* Ethernet de distintas velocidades, economía, capacidad de autoconfiguración e independencia del direccionamiento IP. Ethernet, inicialmente a 3 Mbps, ha evolucionado en capacidad desde 10 Mbps a 100 Gbps y de los simples puentes software que unían dos redes locales se ha pasado a los conmutadores de N*40 Gigabit, equipos con un potencial de conmutación equivalente al de los grandes enrutadores, proporcionando en todo momento familias de conmutadores con un coste por puerto y velocidad sin rival**.** Su evolución se acerca a la conocida "Ley de Moore" de duplicación de la capacidad de los dispositivos semiconductores cada 24 meses [Moo65].

Esta evolución está planteando cambios importantes en la arquitectura de las redes locales, metropolitanas y de proveedor y, más recientemente y de forma muy drástica, en las redes internas de los centros de datos, sujetos a exigencias de tamaño, prestaciones, coste y consumo de energía nunca vistas hasta el momento. Los principales retos a los que se enfrenta la tecnología al ampliar no solo el tamaño de las redes sino también las velocidades de trabajo y el entorno de servicios de aplicación se relacionan fundamentalmente con la necesidad de proporcionar garantías suficientes de escalabilidad, de rendimiento y de seguridad, manteniendo al mismo tiempo sus características de bajo coste, mínima gestión y mínima configuración.

Los puentes transparentes con aprendizaje constituyen el elemento básico en las redes Ethernet conmutadas. Se trata del elemento clave de la tecnología, encargado del proceso de conmutación, que permite alcanzar los destinos deseados dentro de la red, de una manera eficiente, sin necesidad de configurar tablas de encaminamiento ni protocolos de descubrimiento y/o cálculo de rutas (como ocurre en los enrutadores que operan en capa de red). Su evolución, desde sistemas sencillos que servían para interconectar apenas unos pocos segmentos de red, a sistemas de conmutación multipuerto (con enlaces dedicados entre ellos y con cada sistema final) apilables y con capacidad para reenviar cantidades muy elevadas de datos a gran velocidad en redes cada vez más grandes y heterogéneas, es a la vez la clave del éxito de la tecnología y, en cierta medida, su propio talón de Aquiles.

Una vez realizado el primer salto evolutivo que nos lleva a las redes Ethernet conmutadas, tres son los principales factores que emergen como limitantes de la tecnología y que han obligado a sucesivos procesos evolutivos:

a) La utilización de un sistema de direccionamiento plano, sin información topológica que facilite la localización del sistema destino. Las direcciones Ethernet únicas son pre-configuradas en la tarjeta de interfaz de red por el propio fabricante según un esquema de asignación controlada directamente por el propio IEEE. No puede establecerse ninguna correspondencia entre la dirección Ethernet y la localización del sistema que la aloja por lo que se complica el proceso de enrutamiento (sobre todo fuera del entorno de red local de pequeño tamaño en que inicialmente se pensaron utilizar).

b) La utilización del concepto de árbol de expansión como topología lógica de comunicación que permite la difusión de datos en la red sin que se produzcan bucles. La difusión es la única respuesta posible para garantizar la entrega de los datos al destino, cuando el mecanismo de aprendizaje aun no ha tenido oportunidad de localizar una determinada estación destino. Limitar la topología lógica a un árbol sin bucles tiene consecuencias graves en el rendimiento de la misma. Por un lado, se bloquean todos los enlaces que no pertenecen al propio árbol, dejando activos solo un número de enlaces igual al número de nodos de la red menos uno, inutilizando de esta manera una buena parte de la red. Por otro lado, todo el tráfico que deba atravesar la red, para alcanzar nodos situados en otra rama del árbol, debe encaminarse hasta la raíz del mismo para alcanzar la rama correspondiente por donde pueda reenviarse hacia el destino. Esto tiene un doble efecto indeseado: incrementa la cantidad de tráfico en los enlaces cercanos a la raíz (tiende a colapsarlos), e incrementa la distancia lógica entre nodos de la red (produce caminos medios grandes) que topológicamente pueden encontrarse cerca el uno del otro.

c) El recurso último a la difusión como mecanismo de reenvío cuando se desconoce la localización concreta de la estación destino de una trama de datos. En redes de gran tamaño, reducir el alcance de las tormentas de difusión se convierte en una necesidad si se quiere mejorar el rendimiento global de la red.

Pese a que Ethernet es una tecnología normalizada por el grupo de trabajo 802 del IEEE, su éxito y ubiquidad es tal que han surgido innumerables propuestas de mejora, más o menos rompedores con el estándar vigente en cada momento, tanto dentro del propio seno del IEEE como en otros organismos de estandarización como IETF y la comunidad científica en general. En la siguiente sección se revisan sumariamente las principales propuestas surgidas a lo largo de los años primero dentro del IEEE, lo que nos ofrece una panorámica evolutiva del estándar, y luego en otros ámbitos académicos y empresariales.

1.2.1 El estándar IEEE 802.1

La primera versión del estándar Ethernet normalizada por el IEEE (802.3) tenía el entorno de red local segmentada como ámbito de aplicación. Se trataba de redes pequeñas basadas en medios compartidos interconectados mediante puentes con aprendizaje de direcciones MAC (IEEE 802.1D). El principal problema que enfrentaban tenía que ver con la gestión del acceso de la estaciones al medio compartido de transmisión. Para garantizar la difusión de tramas de datos sin bucles se utilizaba el algoritmo del árbol de expansión STP (*Spanning Tree Algorithm*) que transforma la topología física de interconexión de segmentos en un árbol lógico simple, bloqueando todos los caminos ajenos al árbol. Para garantizar un uso eficiente de la infraestructura de red, el diseño de las instalaciones tendía a arborificar la topología real y se utilizaban enrutadores para separar distintos dominios de colisión y difusión, acotando de esta manera el alcance de las tormentas de difusión pero aumentando el coste de configuración y operación de la red global. Además, para reducir el elevado tiempo de convergencia, de casi un minuto, en la reconfiguración del árbol lógico de la topología ante posibles fallos, se tendía a inhabilitar el propio algoritmo STP o a prefijarlo mediante configuración.

La transformación de redes segmentadas a redes conmutadas donde predominaban los enlaces dedicados por estación (dúplex punto a punto), el paralelo y paulatino aumento de la velocidad de transmisión hasta los 100 Mbps primero,(1 y 10 Gbps después), y el vertiginoso aumento en el número de estaciones (tamaño de las redes) provoca en cierta medida la actualización del estándar con la sustitución de STP (principal factor limitante) por un nuevo algoritmo de árbol de expansión RSTP (*Rapid Spanning Tree Protocol*, 802.1w) que permite reducir los tiempos de convergencia de manera drástica (salvo en los casos de caída de la raíz del árbol). Los nuevos conmutadores permiten el reenvío de grandes cantidades de información, y son capaces de almacenar y manejar tablas de reenvío suficientemente grandes para permitir la implantación de grandes redes que operan en capa de enlace, sin apenas configuración; mientras, se sigue manteniendo el uso de enrutadores cuando se desea limitar el alcance de las difusiones y/o separar tráficos por razones administrativas y de tamaño del dominio de fallo en caso de tormentas de tramas. RSTP se incorpora en la versión del estándar 802.1D-2004.

Con el aumento del tamaño de las redes conmutadas, la capacidad de los diseñadores para seguir el modelo topológico en árbol se complica agudizando el problema de rendimiento que se deriva del uso de una topología lógica de transmisión basada en árbol (y el consiguiente bloqueo del resto de enlaces, todos menos el número de conmutadores menos uno). Además, las necesidades administrativas de separación de distintos tipos de tráfico son cada vez más difíciles de canalizar por vías topológicas por lo que se plantean nuevas incorporaciones al estándar que palien dichos problemas. Por un lado, la utilización de etiquetas que permitan clasificar y multiplexar en un mismo enlace distintos tipos de tráficos, dando lugar al concepto de redes locales virtuales (VLAN) recogidas en el estándar IEEE 802.1Q. Por otro lado, la posibilidad de definir múltiples árboles de expansión, diferenciados mediante etiquetas virtuales, que oportunamente implementados permiten aumentar el rendimiento global de la red al reconducir distintos tráficos por topologías lógicas diferentes por lo que no es necesario bloquear ningún enlace de la red. Se trata del estándar 802.1s MSTP (*Multiple Spanning Tree Protocol*) que posteriormente se incorpora a la norma 802.1Q. Su principal inconveniente es la necesidad de configuración manual de las distintas regiones (árboles lógicos) a la vez que siguen necesitando un árbol de expansión común a toda la topología para permitir la interconexión entre redes virtuales.

Del año 2005 en adelante, los esfuerzos de evolución (mas allá de los continuos aumentos de las velocidades de transmisión y conmutación) se concentran en la aplicación de Ethernet al entorno de redes metropolitanas, redes de proveedor y nubes virtuales. En esta escala, nos encontramos con topologías físicas completamente heterogéneas (redes en malla o *mesh*) donde la configuración manual de regiones (árboles) es simplemente inmanejable. Se trata de redes tradicionalmente enrutadas por lo que se opta por buscar una nueva evolución del estándar que incorpore los protocolos tradicionales de enrutamiento (del tipo estado de los enlaces como IS-IS [RFC6329]) pero aplicados a la capa dos. El nuevo estándar, denominado SPB (*Shortest Path Bridging*, 802.1aq), acaba de ser aprobado en marzo pasado.

SPB define un núcleo de red formado por conmutadores cuyo plano de control ejecuta una extensión del protocolo IS-IS para diseminar la información topológica necesaria para el posterior cálculo de rutas mediante un algoritmo de camino más corto como Dijkstra. Dentro del núcleo SPB se calculan árboles de difusión enraizados en cada uno de los nodos y se garantiza la congruencia de las caminos (ramas de los dos árboles correspondientes) que unen dos nodos determinados. Las tramas que acceden a este núcleo SPB son encapsuladas mediante un esquema MAC-in-MAC (802.1ah) dando lugar al modo de funcionamiento denominado SPB-M (por tanto, sin aprendizaje transparente en los nodos que forman el núcleo, por lo que los nodos deben anunciar a qué sistemas finales sirven), o etiquetadas mediante etiquetas de red local virtual (802.1Q/802.1ad) que se asignan al árbol de difusión de cada nodo dando lugar al modo de funcionamiento denominado SPB-V (con aprendizaje transparente). SPB puede combinarse con ECMP [RFC2992] (*Equal Cost Multi-Path*) para

permitir el uso de múltiples caminos y alcanzar de esta manera un uso eficiente de la topología física). Echando la vista atrás podríamos decir que lo que hoy se llama Ethernet ha quedado reducido a poco más que un formato de trama común junto al esquema de direccionamiento MAC clásico (al menos desde la óptica SPB-M).

1.2.2 Otras propuestas

Son innumerables las variantes sobre la arquitectura y los protocolos de Ethernet que se han publicado a lo largo de estos años. Una gran parte, se han centrado en el propio estado del estándar activo en ese momento e incluso, podríamos afirmar que la propia evolución del estándar ha venido influenciada por algunas de ellas.

Otras propuestas, sin embargo, han partido de enfoques divergentes que han puesto en tela de juicio tanto el esquema de direccionamiento (por ejemplo planteando el uso de direcciones jerárquicas) como el uso de árboles de expansión (simples o múltiples) como garantía de difusión libre de bucles (por ejemplo utilizando el esquema de enrutamiento arriba/abajo (*Up/Down routing*), sustituyendo el aprendizaje transparente (por ejemplo por sistemas centralizados, sistemas de registro y traducción de direcciones mediante DHT (*Distributed Hash Tables*) o la aplicación de protocolos de enrutamiento en capa dos) o reduciendo (o eliminando por completo) la difusión de tramas de datos en la red.

De todas ellas, por tratarse de una de las más modernas y de haberse desarrollado en el seno de otro organismo de normalización como es IETF (*Internet Engineering Task Force*) cabe destacar la tecnología TRILL-Routing Bridges (*Transparent Interconnection of Lots of Links*). Se trata de un desarrollo cuyo devenir ha sido bastante paralelo al de SPB y cuyos planteamientos son también muy similares: aplicar el protocolo de enrutamiento de estado de los enlaces IS-IS al cálculo de caminos en capa dos.

En el capítulo dedicado al estado del arte se detallan las propuestas más importantes y se presenta una aproximación a la genealogía y la taxonomía de las mismas.

1.3 Definición del problema y objetivos

El objetivo planteado en el desarrollo de esta Tesis Doctoral consiste en definir protocolos para conmutadores Ethernet avanzados que permitan superar las limitaciones de rendimiento que plantean los entornos de aplicación actuales de la tecnología Ethernet como son: las grandes redes campus, las redes metropolitanas y de proveedor y las redes de centros de datos, y que pueden resumirse en la necesidad clara de una subred IP única, de alta utilización de infraestructura, baja o nula configuración y escalable.

El enfoque de las distintas contribuciones que se realizan es completamente transversal a la línea evolutiva del estándar para posibilitar soluciones más novedosas, cualitativamente distintas y de mayor proyección futura. Para garantizar que se aprovechan las excelentes propiedades en términos de precio/velocidad de la tecnología de transmisión Ethernet se fijó como único (y muy exigente) requisito de partida mantener tanto el formato de trama estándar como el formato de direcciones MAC, decisión que se ha visto avalada por los últimos añadidos al estándar. Además, se plantean como objetivos adicionales reducir en la medida de lo posible la necesidad de configuración (se buscan soluciones distribuidas y auto-configurables), la transparencia con respecto a los estaciones de usuario y la coexistencia con secciones de red basadas en el estándar.

A modo de resumen, los principales aspectos que se exploran en las distintas propuestas presentadas en la Tesis son:

1) La sustitución del árbol de expansión como mecanismo para controlar la difusión sin bucles.

2) La utilización de esquemas de direccionamiento que aporten información topológica y ayuden de esta manera a un encaminamiento más eficiente, siempre dentro de lo permitido por el esquema de direccionamiento MAC.

3) La sustitución o evolución del mecanismo de aprendizaje transparente por/hacia otros sistemas de reenvío que permitan aprovechar al máximo los recursos de la red.

Cada una de las tres principales contribuciones propuestas en esta Tesis se apoya en la combinación de varios de estos aspectos. En realidad, podrían entenderse como una única línea evolutiva en la que cada nueva propuesta se apoya y/o realimenta de los resultados obtenidos en la anterior, en la búsqueda de mecanismos de enrutamiento en capa dos alternativos al esquema dominante (derivado de la aplicación del protocolo IS-IS y el paradigma del enrutamiento por estado de los enlaces).

Inicialmente, se plantea la sustitución del árbol de expansión, como principal factor limitante del rendimiento, por un nuevo mecanismo de difusión libre de bucles basado en la aplicación del esquema denominado enrutamiento arriba/abajo (*Up/Down routing*), o de prohibición de giros, combinado con un esquema de cálculo de rutas tipo vector de distancias (como en RIP). La aplicación de *Up/Down* exige identificar los nodos de la topología mediante un esquema jerárquico (similar a los árboles de expansión) que permite conocer, dados sus identificadores, la posición relativa de los nodos correspondientes dentro de la jerarquía. De esta manera, el protocolo de encaminamiento permite calcular las mejores rutas usando toda la topología física (o casi, up/down prohíbe determinadas combinaciones (giros) de enlaces entrada-salida, pero nunca bloquea enlaces) a la vez que se garantiza la difusión libre de bucles). Se incorpora el esquema de direcciones HLMAC (direcciones locales, jerárquicas codificadas conforme al estándar de direcciones MAC) dando lugar a la arquitectura HURBA (*Hierarchical Up/Down Routing*) y al protocolo HURP. La contribución de esta Tesis a HURP/HURBA consiste en el diseño e implementación de un simulador global de rendimiento de la arquitectura que hace posible su evaluación detallada en una gran variedad de topologías de red.

Posteriormente, se explora la idea complementaria, mantener el árbol de expansión como topología lógica base de enrutamiento y se dota al sistema de la capacidad para utilizar los enlaces que quedarían bloqueados en el estándar R/STP para mejorar el rendimiento del sistema (los denominados atajos). Para conseguir esquivar el árbol de expansión sin incurrir en problemas de generación de bucles, se plantea la aplicación del esquema TRAIN (*Tree Based Routing Architecture for Irregular Networks)* dando lugar a la familia de protocolos TRE, TRE+ y D-TRE (*TRE, Tree-based Routing Ethernet y D-TRE, Dynamic TRE*). Nuevamente, se requiere la utilización de un mecanismo de direccionamiento jerárquico (HLMAC) conjugado con distintos esquemas de adquisición de conocimiento topológico para aplicar al cálculo de rutas, desde el simple reconocimiento de vecinos en TRE, al intercambio controlado de información mediante esquemas vector-distancia (TRE+) o estado de los enlaces (D-TRE). El trabajo realizado en esta Tesis contribuyó al planteamiento inicial de los principios de operación de TRE y su posterior análisis de rendimiento y planteó el camino de evolución hacia TRE+ y D-TRE definiendo los esquemas de enrutamiento y/o protocolos de control necesarios.

Finalmente, la tercera y cuarta contribuciones principales se enmarcan dentro de la línea de investigación actual del grupo que dirige el tutor de la Tesis y que se basa en el desarrollo de una nueva familia de protocolos de enrutamiento en capa dos denominada All-Path (que permiten el uso

de todos los enlaces físicos de la red, sin bloquear ninguno como en STP). En esta Tesis se presenta el primer protocolo de la familia, denominado ARP-Path, al que se ha contribuido desde sus inicios en los que se exploraba la factibilidad de la inundación total (sin bloquear enlaces) de las tramas de difusión en una red sin que se produjeran bucles. Se detalla el mecanismo de control de bucles diseñado para sustituir al árbol de expansión (cuya patente acaba de ser reconocida al grupo) y se presenta el entorno de caracterización del rendimiento y los resultados obtenidos en su evaluación. ARP-Path aprovecha la propia difusión de las tramas de control del protocolo ARP (en concreto la trama *ARP-Request*) para detectar sobre la marcha el camino de menor latencia que existe en la red, en ese momento, entre dos sistemas origen y destino que quieren intercambiar tráfico. El mecanismo de difusión controlado que se ha diseñado permite explorar todos los caminos posibles y detectar de entre ellos el de menor latencia (por tanto, es un enrutamiento de tipo camino más corto) garantizando a la vez que no se producen bucles en el proceso. De esta manera, se elimina la necesidad de aprendizaje. Podríamos decir que ARP-Path es un protocolo que opera en el plano de datos, al no requerir un protocolo de control específico (salvo para acelerar la reconstrucción de caminos tras fallos en la red). Un elemento clave desarrollado en esta contribución es el simulador de flujos de ARP-Path que permite caracterizar su comportamiento con el suficiente nivel de abstracción y requisitos temporales y computacionales asequibles y que está permitiendo mostrar la sorprendente capacidad nativa de reparto de carga del protocolo ARP Path, derivada de establecer los caminos cuando se precisan y en función de la latencia instantánea de la red.

1.4 Estructura de la Memoria

A fin de facilitar la lectura de la Tesis y dar una visión general de la misma, se describe a continuación su estructura y el contenido de los capítulos en los que se dividido. La Tesis está estructurada de la forma siguiente: en el Capítulo 1 se ha presentado el contexto actual de las redes campus y de centros de datos Ethernet, su evolución, tendencias y carencias actuales; se ha adelantado una definición del problema a resolver y de los requisitos. En el Capítulo 2 se analiza el estado del arte en las áreas directamente relacionadas y en otras áreas colaterales, exploradas por presentar aplicabilidad potencial de conceptos. En el Capítulo 3 se describen los protocolos de prohibición de giros (HURP) propuestos y las simulaciones realizadas. En el Capítulo 4 se describen los protocolos jerárquicos y con direccionamiento topológico propuestos (TRE, TRE+, DTRE1 y DTRE2) y una comparación detallada de sus prestaciones. En el Capítulo 5 se describen las contribuciones realizadas a la nueva familia de protocolos de baja latencia ARP-Path y a su evaluación mediante un generador de flujos. Finalmente, en el Capítulo 6 se elaboran las principales conclusiones y se describen las orientaciones para el trabajo futuro.

2 Estado del arte

En este capítulo se revisa el estado del arte de los Conmutadores de Funcionalidad Añadida (CFA) también denominados Conmutadores Ethernet Avanzados, (CEA) o *Shortest Path Bridges* (entendidos de forma genérica) en los aspectos relacionados con los objetivos de la Tesis.

Las diversas propuestas de CFA realizadas a lo largo del tiempo hacen distinto énfasis en la resolución de los diversos problemas de las redes conmutadas actuales y se centran en escenarios de red de variado tamaño y problemática, desde las redes campus a los centros de datos y redes metropolitanas. Por ello los aspectos concretos que cada propuesta intenta mejorar son múltiples y su importancia es muy variable en los distintos escenarios. Dichos aspectos son: escalabilidad, limitación de la difusión de tramas, grado de utilización de la infraestructura, encaminamiento y separación de tráficos. Por ello y para una mejor comprensión del estado del arte de las propuestas de CFA se ha realizado una clasificación de las propuestas existentes que facilita su comprensión. No se describen las propuestas ligadas a topologías específicas de centros de datos por ser de enfoque muy especializado en una topología determinada, su desarrollo es muy reciente y no son objeto de la presente Tesis.

El capítulo se estructura de la siguiente forma: se describen primero las funciones y mecanismos básicos de los puentes de funcionalidad avanzada y una panorámica de la evolución conceptual de las propuestas de puentes transparentes, a continuación la clasificación de las propuestas y se describe y valora sucintamente cada una de ellas.

2.1 Funciones y mecanismos de los puentes

Describimos aquí las funciones de los puentes en las arquitecturas de capa dos que emplean Conmutadores de Funcionalidad Añadida (CFA). La Ilustración 1 muestra el escenario de referencia para los CFA con los elementos que intervienen en el reenvío de tramas: sistemas finales (hosts), puentes estándar (*Standard bridge*), puentes de funcionalidad añadida localizados en los bordes de la red (E) y CFA situados en el núcleo de la red (C). Los enrutadores, no mostrados en la figura, están localizados y marcan los límites de la red de capa dos, de forma que la red conmutada es una subred IP única. Los puentes estándar pueden estar situados en cualquier parte, entremezclados con CFA o con algunas restricciones, por ejemplo, solamente pueden estar fuera del núcleo de la red, en la forma denominada como núcleo-islas (*core-island mode).* Los servidores de localización (LS) son elementos opcionales responsables de almacenar la localización de los sistemas finales (SF, hosts) o resolver las direcciones IP (ARP Proxy) y se utilizan para evitar o reducir las tramas de difusión y mejorar así la escalabilidad.

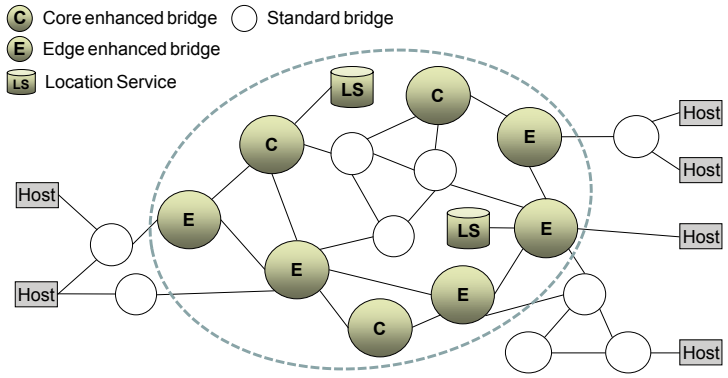


Ilustración 1. Escenario de referencia de red con puentes de funcionalidad añadida

2.1.1 Identificación de Puente y de Sistema Final

Los puentes CFA reenvían tramas a la dirección contenida en la trama Ethernet en el campo de dirección destino, la cual puede corresponder a un puente, a un enrutador o a un SF y en algunos casos, según el contenido de otros campos como la etiqueta VLAN. El identificador de SF puede tener el formato de una dirección MAC estándar (global o local). Las direcciones estándar globales MAC se caracterizan por ser asignadas en fábrica, globalmente únicas y planas, sin jerarquía. Contienen un identificador OUI (*Organizational Unique Identifier*) que identifica al fabricante, además de un número asignado consecutivamente en los tres octetos restantes. Las direcciones MAC asignadas localmente, por el contrario, pueden ser estructuradas e incluso tener significado topológico para facilitar la agregación de rutas. Los identificadores estructurados de SF pueden incluir un identificador de puente, que puede tener la forma de un prefijo, en el que el identificador de SF hace de sufijo de dirección. La asignación de este tipo de identificadores jerárquicos requiere configuración manual o protocolos específicos de asignación de direcciones, que deben ejecutarse antes de que pueda procederse al reenvío de tramas.

2.1.2 Localización de SF

Para reenviar tramas a un SF determinado, los puentes deben previamente obtener la posición del SF destino, lo que puede realizarse mediante un sistema de difusión y aprendizaje hacia atrás, de registro y consulta en un servidor de localización centralizado o distribuido, o simplemente

descodificando paso a paso la información topológica contenida en la dirección destino, si existiera. Un SF puede registrarse a sí mismo, lo que requiere modificar el software de los sistemas finales, o bien puede ser registrado por su puente designado, el puente responsable del reenvío de sus tramas. El puente designado puede asimismo incluir la función de *proxy* ARP. En estos casos cada puente es responsable de recolectar las direcciones MAC e IP de sus SF directamente conectados y de hacer disponible esta información al resto de los puentes. La localización del SF es a menudo identificada por su puente designado o por el puerto de salida para alcanzarlo. Los mecanismos para almacenar y distribuir esta información (resolución del puente destino) pueden tener la forma de tablas hash distribuidas (DHT [Jai92]), bases de datos replicadas, o utilizar simplemente distribución multicast a todos los puentes.

2.1.3 Reenvío de tramas

El reenvío de tramas en un puente consiste principalmente en determinar el puerto de salida de la trama recibida. Esta decisión se basa en direcciones de trama y en información de estado almacenada en el puente. Se pueden utilizar identificadores adicionales tales como etiquetas de VLAN para filtrar el aprendizaje de direcciones MAC y éstos pueden ser combinados con direcciones destino para tomar las decisiones de reenvío.

Los mecanismos básicos para CFA son la difusión limitada de tramas (con o sin aprendizaje hacia atrás) y el encaminamiento unicast. La difusión total de tramas normalmente se limita con el fin de evitar bucles y las consiguientes tormentas de tramas. Así ocurre con los mecanismos de prohibición de enlaces (la difusión se permite solamente a través de uno o de múltiples árboles de expansión pero disjuntos, separados por pertenencia a VLAN o conjuntos de VLANs distintos). Alternativamente, los mecanismos de prohibición de giros, prohíben ciertas combinaciones de puertos de entrada y de salida en el reenvío y no deshabilitan (prohíben) enlaces.

El encaminamiento puede ser centralizado o distribuido. El encaminamiento centralizado puede adoptar múltiples variantes: puede ser calculado automáticamente tras un proceso de adquisición de topología y luego configurado en cada puente, por el protocolo o por una aplicación externa que realiza el cálculo de las rutas y configura los puentes mediante mecanismos de gestión de red. En el encaminamiento distribuido, cada puente calcula autónomamente sus rutas. El cálculo de caminos mínimos a un destino determinado puede realizarse mediante intercambio de información de encaminamiento tipo Vector Distancias o de Estado de Enlaces y aplicando algoritmos como los de Bellman-Ford o Dijkstra, respectivamente. Los árboles de caminos mínimos se utilizan para la obtención de rutas unicast y multicast. La construcción de estos arboles comprende dos operaciones básicas: elección del puente raíz y cálculo del árbol. La elección del puente raíz en muchos casos viene predeterminada si se calcula una instancia de árbol por cada puente de la red. El cálculo de estos árboles puede hacerse mediante protocolos de árbol de expansión o mediante protocolos de encaminamiento.

2.2 Mecanismos de control de bucles

Un aspecto crucial en los puentes transparentes es la prevención de bucles de tramas. El control de bucles es más complejo en la capa de enlace que en la de red, por la ausencia de un campo TTL (*Time-to-Live*) en la trama Ethernet que permita descartar de forma simple las tramas extraviadas en la red. El uso extensivo de la difusión de tramas y replicación de las tramas con destino desconocido por los puentes agrava el riesgo de bucles y aumenta la posibilidad de tormentas de tramas las cuales pueden dejar la red colapsada durante períodos significativos por la sobrecarga de tramas en bucle.

Podemos clasificar los mecanismos de control de bucles en dos tipos: prevención de bucles y mitigación de bucles. La categoría de prevención de bucles incluye todos los mecanismos que previenen completamente la aparición de bucles incluso de forma transitoria. Estos mecanismos utilizan normalmente el plano de control para crear estado y/o identificadores que son utilizados para imponer en el plano de datos algunas restricciones al reenvío de tramas. Por ejemplo, los protocolos de árbol de expansión como STP y RSTP construyen árboles de enlaces activos y bloquean todos los enlaces redundantes (los que excedan el número de nodos menos uno) de la topología. Estos se categorizan como mecanismos de prohibición de enlaces. Sin embargo estos protocolos pueden sufrir bucles de tramas en ciertas circunstancias tales como configuración errónea de los enlaces (por ejemplo, un extremo de un enlace configurado como full dúplex incondicional y el otro como semidúplex) que llevan al puente a operar en el modo de inundación completa. Los mecanismos de tipo protección en anillo incluyen las propuestas basadas en prohibición de enlaces específicas para topologías en anillo. Al limitarse las topologías a la de anillo el protocolo se simplifica mucho y se obtiene reconfiguración rápida porque bloqueando solamente un enlace se evitan los bucles en toda la red.

Una alternativa menos restrictiva que la de prohibición de enlaces es la de prohibición de giros, término genérico que incluye a las propuestas que evitan los bucles prohibiendo ciertas combinaciones (denominadas giros) de puertos de entrada y puertos de salida en los puentes. El cálculo de los giros a prohibir en cada nodo, para evitar los bucles, puede realizarse de forma centralizada o distribuida, pero normalmente tras cierta coordinación central a fin de asegurar la coordinación en la red.

La categoría de mitigación de bucles incluye todos los mecanismos que reducen los efectos de los bucles pero no previenen su aparición. Su objetivo es limitar el impacto de los bucles transitorios que pueden aparecer típicamente debido a la inconsistencia transitoria de las tablas de reenvío de los puentes hasta que se alcance la sincronización. Estos son un complemento de los mecanismos de prevención. Es el caso del campo TTL y de mecanismos propietarios que limitan el ancho de banda para cada tráfico en caso de bucle, para evitar la caída total de la red en caso de tormenta de tramas. La Ilustración 2 muestra una clasificación de los mecanismos. Como puede verse en algunos protocolos están estrechamente vinculados el método de enrutamiento y la prevención de bucles. En secciones posteriores se estudian más en detalle dichos mecanismos.

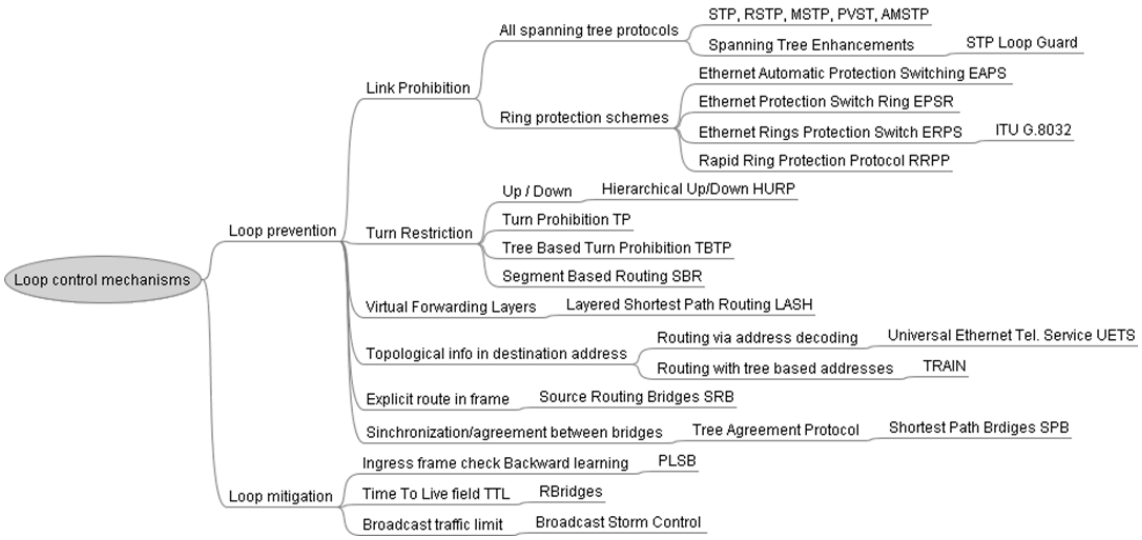


Ilustración 2. Clasificación de los mecanismos de control de bucles en puentes transparentes

2.3 Identificación, separación y agregación de tráficos

Además de las direcciones de SF y de puente, algunas etiquetas, como la de VLAN y sus derivaciones y extensiones tales como las de VLAN de Cliente y de Servicio (*Customer* VLAN, C-VLAN y *Service* VLAN, S-VLAN), son utilizadas también por el mecanismo de reenvío para identificar, separar y agregar flujos de usuarios y/o servicios.

La estandarización en IEEE 802.1Q de las VLAN hizo posible la creación de dominios de difusión separados y de aislar dominios en fallo, mejorando las prestaciones (especialmente la utilización de la infraestructura) y la seguridad. Pero el límite de 4095 valores para el identificador de VLAN (12 bits) planteaba una seria limitación para que un proveedor de red preste servicios de conexión Ethernet entre redes de un mismo cliente situados en diferentes sitios a muchos clientes diversos. El mecanismo para introducir jerarquía en las VLANs fue el apilamiento de VLAN, también denominado Q-in-Q, en el que en la frontera de red del proveedor se inserta delante la etiqueta VLAN del proveedor, que será la utilizada para el reenvío en la red del proveedor, independizándola así de las etiquetas VLAN asignadas por el cliente. Este mecanismo permite la separación lógica entre cliente y servicio y evita la necesidad de coordinación entre clientes y proveedores del servicio de interconexión. Pero este mecanismo no es suficiente para redes inter-proveedor, por lo que a estos campos, se añadieron otros identificadores como el campo I-SID en SPBM, para identificar de forma precisa el servicio en los puertos de la frontera de la red, y el apilamiento de las direcciones MAC para separar el reenvío en la red del proveedor, haciéndolo depender de las direcciones de puente y ocultando las direcciones MAC (de gran número) de los usuarios.

2.3.1 Funciones de adaptación para coexistencia en redes híbridas (con redes *legacy*)

Un requisito esencial de los puentes CFA es su coexistencia e interoperabilidad con puentes estándar IEEE 802.1D y 802.1Q. En las redes híbridas deben coexistir ambos, cada uno con sus mecanismos de reenvío específicos. Los identificadores de puente, de SF y de protocolo (campo *Ethertype*) son los utilizados para saber si la trama debe ser procesada en el modo estándar o en el modo 'mejorado'. Un escenario frecuente, incluso en los protocolos IEEE estándar, consiste en realizar una cierta adaptación de la trama en los puentes frontera, al entrar en la red de puentes CFA. Los puentes frontera deben realizar todo el procesado 'mejorado' CFA de la trama y luego reenviarla a los puentes del núcleo hasta alcanzar el/los puente(s) frontera de salida donde se realiza la adaptación inversa de la trama. Las funciones pueden incluir el registro del SF en un servidor de localización y la inserción/extracción o sustitución de identificadores de reenvío.

A diferencia de los puentes estándar, que operan con aprendizaje de direcciones de SF, los CFA pueden usar identificadores especiales para transportar tramas en la red mediante encaminamiento entre los puentes CFA. Se utilizan tres tipos de mecanismos para facilitar esta identificación de la trama: encapsulado, traducción de direcciones y delimitación de rangos de los identificadores. El encapsulado de las tramas estándar Ethernet sobre tramas mejoradas permite su transporte sobre el dominio de puentes CFA. La cabecera de encapsulado puede añadir campos extra para incluir identificadores adicionales para la identificación del cliente o del servicio y un campo TTL o de número de secuencia de fragmento de trama. La traducción (sustitución) de direcciones en los puentes de entrada y salida de la red CFA puede realizar la misma funcionalidad que un túnel. Esta traducción puede tomar la forma de NAT de MAC, reemplazando la MAC global del SF por una dirección asignada por el puente frontera de entrada.

Otro mecanismo para identificar la necesidad de procesado específico de la trama consiste en reservar un rango de un identificador para las tramas con procesado especial. Es el caso de las VLAN IDs, utilizadas por *Provider Backbone Transport* [8021ah][PBBTE] para excluir una serie de VLAN IDs de la conmutación del puente mediante el protocolo MSTP u otro, y ser controladas mediante conexiones estáticas configuradas mediante un sistema de gestión centralizado. Igualmente es el caso de las propuestas que utilizan direcciones MAC locales para las tramas de procesado CFA, en las que el bit U/L las distingue de las direcciones MAC globales, coexistiendo con las tramas con direcciones MAC globales, que reciben tratamiento de puente estándar por el puente CFA.

2.4 Clasificación de los puentes

Antes de presentar la evolución de los puentes transparentes y para proporcionar una visión más clara de las propuestas de puentes desde un punto de vista funcional, en la Ilustración 3 proponemos una clasificación de los puentes presentada en [ICG+09]. Una primera división de los puentes los divide en puentes basados en árboles y puentes enrutadores. Los primeros se subdividen en árbol único o múltiple. Los de árbol único a su vez comprenden los de árbol estándar como STP/RSTP[8021D98][8021D04] y los que incluyen el uso de enlaces cruzados como STAR [Lui02][LLN02] o TRAIN [CT97]. Los de árboles múltiples comprenden los de árboles no apilados y los apilados. Dentro de los árboles no apilados se incluyen los que utilizan arboles basados en etiquetas VLAN, categoría a la que pertenecen muchos de los protocolos descritos: MSTP [8021Q03], PVST [PVST], Viking [Sha+04] y SPB [8021aq] (en su variante SPBV). Diferenciamos en esta categoría entre instancias de árbol enraizadas (SPB) o arbitrarias (MSTP). El resto, sin etiqueta VLAN, es una categoría marginal que solamente incluye dos propuestas de escasa repercusión. Los de etiquetas VLAN anidadas incluyen a Q-in-Q [8021ad] y GOE [Iwa+04]. Dentro de la categoría con árboles apilados distinguimos según sean árboles simples o múltiples los apilados, destacando las propuestas en esta segunda categoría, debidas a las mayores prestaciones de los árboles múltiples y su utilidad para escalar las redes a grandes tamaños. Destacan aquí PBB [8021ah], PVT [KS06], PLSB [8021aq] y PBB-TE [PBBTE] como predecesores del actual estándar SPB en su variante SPBM (encapsulado MAC in MAC).

Los segundos, puentes enrutadores, se clasifican según el encaminamiento sea centralizado o distribuido. Los primeros pertenecen mayoritariamente a la categoría de prohibición de giros descrita más arriba, mecanismo que se beneficia del conocimiento centralizado de la topología. Los segundos se dividen en proactivos y reactivos. Los reactivos incluyen solamente los de encaminamiento en origen, pudiendo incluirse en esa categoría la familia de protocolos *All-Path* a la que pertenece el protocolo ARP-Path recientemente desarrollado en el grupo GIST-Netserv y sus variantes *Flow-Path* y *Tree-Path*. Los que utilizan encaminamiento proactivo se clasifican en vector distancia, al que pertenece HURP [Iba+08b][IGC+10], descrito en la presente Tesis, estado de enlaces, con el mayor número de propuestas y las de mayor impacto (RBridges [Per04], SPB, SEATTLE [KCR08]) y, el resto, en otros en los que se incluye Smartbridge [RTA00] y EFR/UETS [MI06], el cual descodifica la dirección jerárquica destino en cada puente para encaminar las tramas, funcionalidad también disponible en HURP.

El reciente interés por las redes de centros de datos está resultando en múltiples propuestas de encaminamiento especializadas y orientadas exclusivamente a topologías de red denominadas *Fat-tree*, a veces correctamente y, a veces, de forma incorrecta como es el caso de Portland [Mys+09], al tratarse realmente de una red Clos plegada. Dentro de estas topologías coexisten los enfoques anteriormente indicados, desde los basados en MSTP con varias VLAN separadas por árboles (un puente de núcleo como raíz de un árbol por VLAN para implementar un plano de conmutación separado, con típicamente cuatro árboles paralelos)

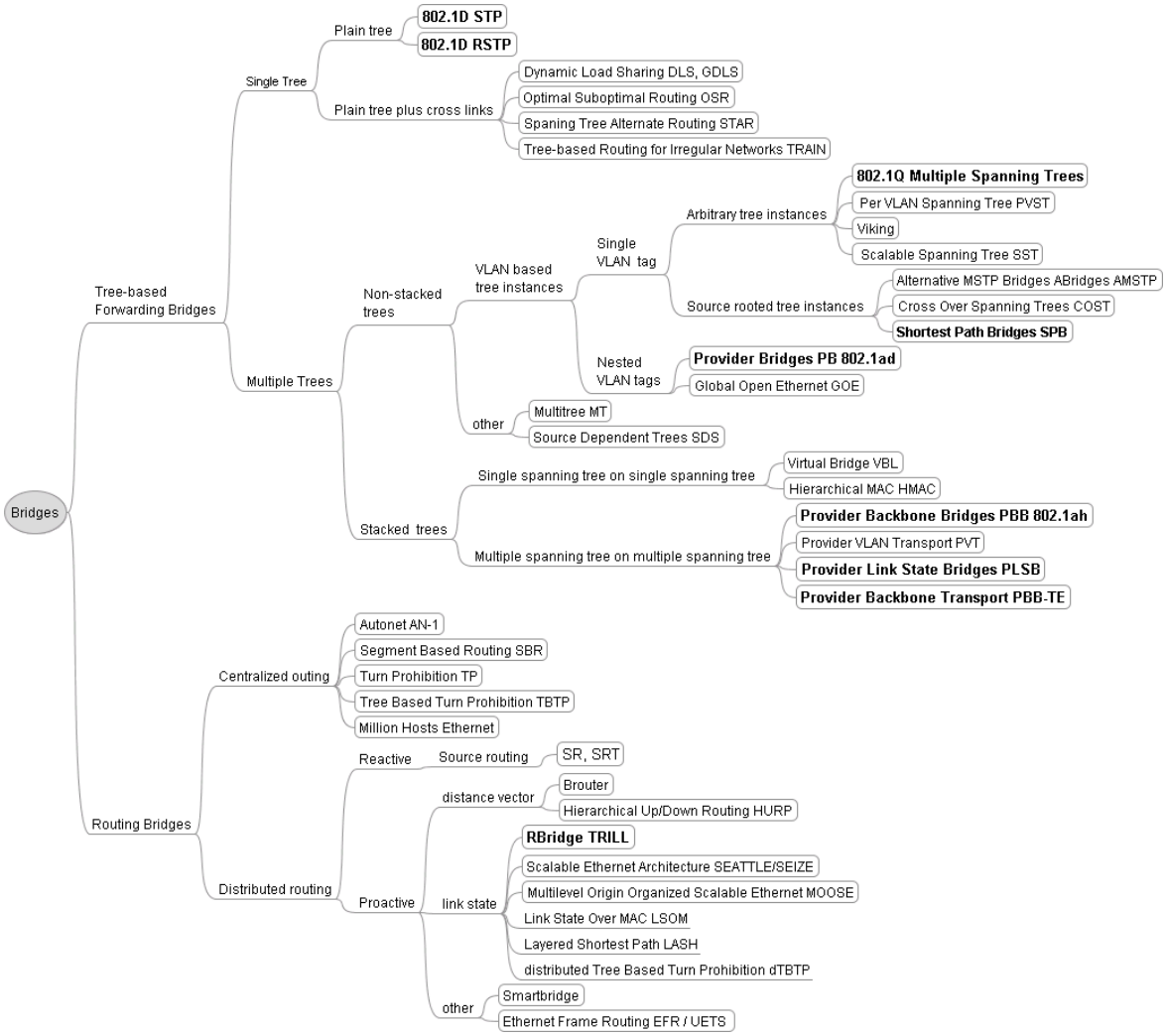


Ilustración 3. Clasificación de los puentes

2.5 Evolución conceptual de los puentes

Describimos ahora sucintamente la evolución temporal de los paradigmas de puentes. El mapa de la Ilustración 4 es una vista conceptual y genealógica de la evolución de las propuestas de puentes agrupadas según los paradigmas básicos empleados. Está ordenada temporalmente de manera aproximada de izquierda a derecha. Las siglas se explican en la Tabla I. Este estudio incluye las propuestas de protocolos aplicables a topologías irregulares de puentes del tipo de almacenamiento y reenvío (*store and forward*). No se incluyen, por tanto, los aplicables solamente a redes de interconexión con *wormhole routing* como Myrinet [BCF+94] y otros. Los principios algorítmicos fundamentales de cada propuesta se muestran mediante el color de fondo y en los lados izquierdo y derecho y su dependencia genealógica se demuestra con las flechas que conectan las burbujas. La figura también demuestra, para cada propuesta, si el protocolo está estandarizado, en proceso de estandarización o es un protocolo propietario y, asimismo, si utiliza encapsulado. Algunos protocolos y dependencias entre ellos no se muestran para preservar la legibilidad de la figura. La Tabla 1 resume los aspectos más destacables de cada propuesta.

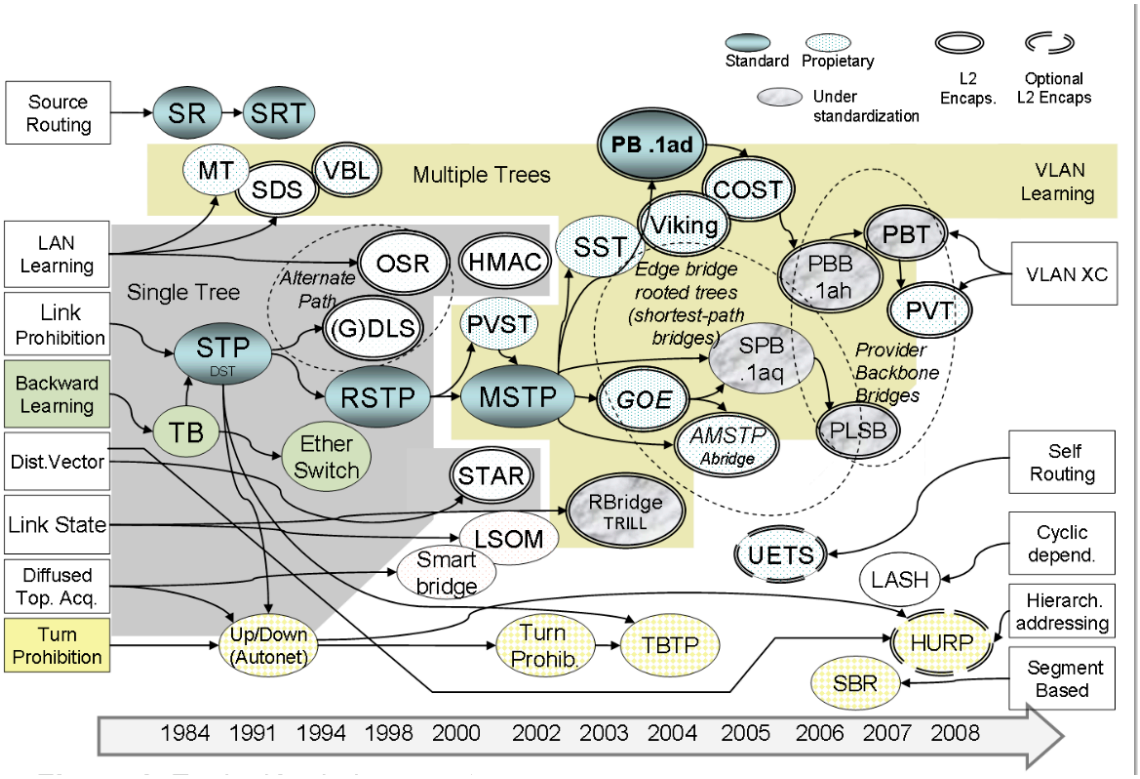


Ilustración 4. Evolución de los paradigmas de puentes

2.6 Puentes de encaminamiento en origen

Los puentes *Source Routing* (SR) [DP88] fueron diseñados por IBM en los años 80 y se estandarizaron dentro del grupo IEEE 802.5 para el encaminamiento en redes *Token Ring* de varios anillos. Los puentes SR insertan la ruta explícita en la trama para encaminar tramas a través de varias LAN y de los puentes que las unen entre sí. Esta ruta es una secuencia pedida de identidades (ID) de LAN y de los puentes que se atravesarán, invirtiendo la orden expresan la ruta en sentido opuesto. No se utiliza protocolo de árbol de expansión. Las rutas se obtienen a través de mecanismos de descubrimiento por inundación. Los puentes SRT (*Source Routing Transparent*) combinan la funcionalidad de los puentes SR y de los puentes transparentes. La Ilustración 5 muestra su posición evolutiva. Sus conceptos de encaminamiento bajo demanda no han tenido éxito en redes cableadas, pero son bastante adoptados en redes inalámbricas en protocolos como *Dynamic Source Routing* (DSR).

Tabla 1. Características principales de las propuestas de puentes

ACRONIMOS	Nombre	Resumen	Estándar / Propietario	Autores	Fecha Publicación	Cómputo	Compatibilidad con hosts	Compatibilidad bridges 802.1D	Encapsulado	Etiquetas usadas
TB	Transparent Bridge	Puente de dos puertos aprend	Propietario	M. Kempf	1984	Aprendizaje de MAC por puerto	S	S	N	N
STP/DEC	Spanning Tree Protocol	Arbol de expansion con temporizadores	Propietario	R.Perlman, Digital Equip. Corp.	1987	Construcción de árbol según distancia a bridge raíz	S	N	N	N
STP/IEEE	Spanning Tree Protocol	Arbol de expansion con temporizadores	Estándar	R.P, IEEE 802.1D	1987	Construcción de árbol según distancia a bridge raíz	S	S	N	N
SRB	Source Routing bridges	Ruta explicita (Bi,LAN,...)	Estándar	IBM	1980's	Ruta bajo demanda por inundacion	N	S	N	Ruta explícita
ES	Ether Switch	Multiport hardware bridge	Estándar	Kalpana Co.	1989	Aprendizaje de MAC por puerto	N	S	N	N
DLS/GDLS	Distributed Load Sharing	Arbol de expansion más enlaces transversales seleccionados	Propietario	Hart/Perlman	1989/ 1992	Retardo relativo enlaces transversales/ árbol	S	S	N	Ethertype
Autonet	Autonet	Encaminamiento Up/Down basado en numeración puentes mediante STP	Propietario	Shoreder/Rodeheffer	1991	Rutas legales sin giros abajo-arriba	S	N	S	Direcciones cortas
VBL	Virtual Bridges	Bridge Virtual que une a islas con árbol de expansión	Propietario	Casale Catania Puliafito	1994	Arbol de expansión simple de nivel superior en el núcleo de la red	N	S	S	N
Smartbridge	Smartbridge	Caminos mínimos sobre arboles basados en origen	Propietario	Rodeheffer	2000	Diffusing computations. Adquisición de topología por los puentes. Tablas de localización de hosts por segmento	Sw	N	N	N
MSTP 802.1Q	Multiple Spanning Tree Protocol	Instancias múltiples de árboles de expansión	Estándar	IEEE 802.1Q	2002	Construcción de árboles múltiples según distancias a bridges raíz de instancias	N	S	N	VLAN
STAR	S.Tree Alternate Routing Protocol	STP más enlaces transversales autodescubiertos	Propietario	K.S. Lui	2002	Rutas adicionales directas descubiertas entre bridges STAR. Estimación distancias	N	S	S	Ethertype
TP, TBTP dTBTP	Turn Prohibition	Evitación de bucles [mas] árbol de expansión	Propietario	Starobinsky, Pellegrini	2002,2004	Cómputo de un conjunto de giros prohibidos del grafo de red que evita todos los bucles	N	S*	N	E
LSOM	Link State Over MAC	Encaminamiento de estado de enlaces entre puentes de dorsal	Propietario	Garcia/Duato	2003	Cómputo de camino minimo a puente frontera	N	N	N	N
RSTP 802.1D	Rapid Spanning Tree Protocol	Mecanismos de convergencia rápida, enlaces punto a punto	Estándar IEEE	Cisco/ IEEE 802.1s 802.1D	2004	Construcción del árbol	N	S	N	N
Rbridges	Routing Bridges	LSP over encapsulated MAC with TTL	Estándar IETF	R Perlman /TRILL	2004	Cálculo de árboles mediante estado de enlaces, listas de hosts difundidas bridges	N	S	S	Ethertype
AMSTP ABridges	Alternative Multiple Spanning Tree Protocol	MSTP simplificado autoconfigurable con arbol autoenraizado en cada bridge sin aprendizaje	Propietario	G. Ibáñez/ A. García	2004	Una instancia de árbol por ABridge (núcleo). Reenvío al ABridge frontera de salida via puerto raíz de cada bridge	N	S*	S	Ethertype
Viking	Virtual LAN Cluster Networking	VLAN y MSTP controlados por aplicación para optimizar rendimiento	Propietario	Sharma, Gopalan	2004	Caminos agrupados sobre instancias de árbol y VLANs compuados y configurados via gestión con caminos de reserva precalculados	S	S	S	VLAN
SSTP	Scalable Spanning Tree	Extension de MSTP para múltiples regiones autodimensionadas	Propietario	Ishizu et al.	2004	Optimizacion automática del tamaño dimension de las regiones	S	S*	N	VLAN
PB 802.1ad	Provider Bridges	VLANs anidadas	Estándar	IEEE	2005	Arboles multiples de expansio	N	S	S	Q-in-Q
UETS	Universal Ethernet Telecom Servicie	Aplicaciones sobre LLC (sin IP) y direcciones MAC jerarquicas locales	Propietario	J. Barroso	20005	Encaminamiento por decodificacion de direccion y tablas de encaminamiento complementarias	Dual stack IP / UETS	S*	Encap. to traverse UETS domain	N
GOE	Global Open Ethernet	Conmutación de etiquetas, L2VPN, anidadas	Propietario	Iwata / NEC	2002/2004	Arboles multiples a nodos frontera	N	S	S	Jerárquica Recursiva
SPB 802.1aq	SHORTEST PATH BRIDGING	MSTP con árboles simétricos enraizados en puente de borde. Aprendizaje MAC compartido entre VLANs	Estándar	IEEE	En curso	Instancias de árbol autoenraizadas calculadas con estado de enlaces	N	S	N	STD .1Q tag
PBT (PBB-TE)	Provider Backbone Transport	Reenvío basado en VLAN+MAC	Estándar	Nortel, BT	En curso	Conexiones configuradas estáticamente	N	S*	S	MAC-in-MAC
PBB 802.1ah	Provider Backbone Bridges	Conexión Interproveedor (MAC in MAC)	Estándar	IEEE	En curso	B-DA and B-SA para direcciones origen y destino de troncal	N	S	S	MAC-in-MAC

2.6.1 Puentes transparentes (TB) y conmutadores

M. Kempf [Kem86] introdujo en 1984 el concepto de puente transparente para interconectar dos segmentos LAN, filtrando el flujo de tráfico entre dos segmentos con el aprendizaje de las direcciones MAC asociadas a cada puerto. El bridge almacena y reenvía la trama recibida de una LAN en la otra LAN, respetando el protocolo CSMA/CD, por lo que actúa como un único host en la LAN destino a efectos de colisiones, reduciéndolas significativamente. Esto posibilitó la interconexión de varias LAN filtrando el tráfico local de las LAN mediante el aprendizaje en cada puerto de las direcciones MAC origen de los hosts emisores de las tramas, evitando el cuello de botella de tráfico en la LAN debido a la conexión de un número creciente de hosts en un único segmento LAN compartido. Eran dispositivos software, (lo que limitaba sus prestaciones) que permitían segmentación e interconexión de LAN, y utilizaban el árbol de expansión o topologías en árbol para evitar bucles.

Los conmutadores (denominados inicialmente como '*Ether Switch*') fueron desarrollados por primera vez por la compañía Kalpana en 1989 como evolución del puente de dos puertos basado en software. Eran puentes transparentes multipuerto implementados en hardware. Su diseño, con un *backplane* de conmutación de alta capacidad, permitió la interconexión no bloqueante de segmentos LAN. La evolución natural de los conmutadores ha ido hacia la generalización de enlaces punto a punto, con lo que la función de conmutación se ha desplazado desde el segmento Ethernet compartido al conmutador. La aparición de los conmutadores fue cercana en el tiempo a la de los enrutadores. En 1986 Cisco lanzó su primer enrutador comercial (AGS).

2.6.2 Árbol de expansión más enlaces cruzados seleccionados

Diversas propuestas aparecieron para mejorar la baja utilización de la infraestructura que presenta STP y permitir incluso el reparto de carga. Estas propuestas se basan en habilitar el uso de algunos enlaces transversales entre puentes, enlaces que no pertenecen al árbol de expansión (*cross links*) y que normalmente son bloqueados por el protocolo STP. Hart [Har88][Har89] propuso DLS con este enfoque. GDLS (*Generalized DLS*) es una extensión y simplificación de la propuesta de Hart para evitar algunos inconvenientes de DLS. Más concretamente, Perlman y otros propusieron en [PHL92] identificar los enlaces no pertenecientes al árbol de forma que puedan usarse como caminos alternativos, más cortos que los del árbol. GDLS compara la velocidad del enlace transversal con la del camino por el árbol.

2.7 Protocolos de árboles múltiples de expansión

Después de la invención del protocolo de árbol de expansión único, el concepto de LAN virtual (VLAN) creó la virtualización de la infraestructura y la posibilidad de separar completamente diversos tráficos dentro de la misma infraestructura física, creando redes virtuales completamente estancas. El protocolo de árbol de expansión multiplicó las instancias de árbol, vinculándolas a diversos grupos de VLAN, a los que pertenecen los puertos y tramas. Numerosas propuestas utilizan este paradigma con diversas variantes como se ve en la Ilustración 3.

2.8 Protocolo de árbol de expansión múltiple

El protocolo *Multiple Spanning Tree Protocol* (MSTP) [8021Q03] ha sido estandarizado en el IEEE 802.1Q. MSTP utiliza un árbol común (*Common Spanning Tree, CST*) que conecta todas las regiones de la topología. En cada región MST se configuran varias instancias de árbol de expansión. Las instancias de árbol se construyen mediante una *Bridge Protocol Data Unit* (BPDU) combinada que contiene datos de todos los árboles de expansión simultáneos en la región. Una instancia interna de RSTP (IST) controla cada región, tiene un bridge raíz regional y la IST forma parte del árbol de expansión (CST) común que engloba a todas las regiones.

2.9 Puentes enrutadores

La obtención de caminos mínimos en redes conmutadas se ha resuelto en muchas de las propuestas utilizando protocolos de encaminamiento de capa de red en la capa de enlace, al tratarse de protocolos muy probados y maduros. De entre ellos predominan los de vectores distancia y de estado de enlaces. La Ilustración 5 muestra parte de la evolución de dichas propuestas y su utilización de protocolos de encaminamiento en capa dos, entre las que se debe incluir *Shortest Path Bridges* recientemente estandarizado. Son mayoritarios los protocolos basados en estado de enlaces, por su mayor velocidad de convergencia, si bien los protocolos de vector distancia son más simples y eficientes cuando se utilizan sólo como complemento para poder aprovechar los enlaces bloqueados por el protocolo de árbol de expansión.

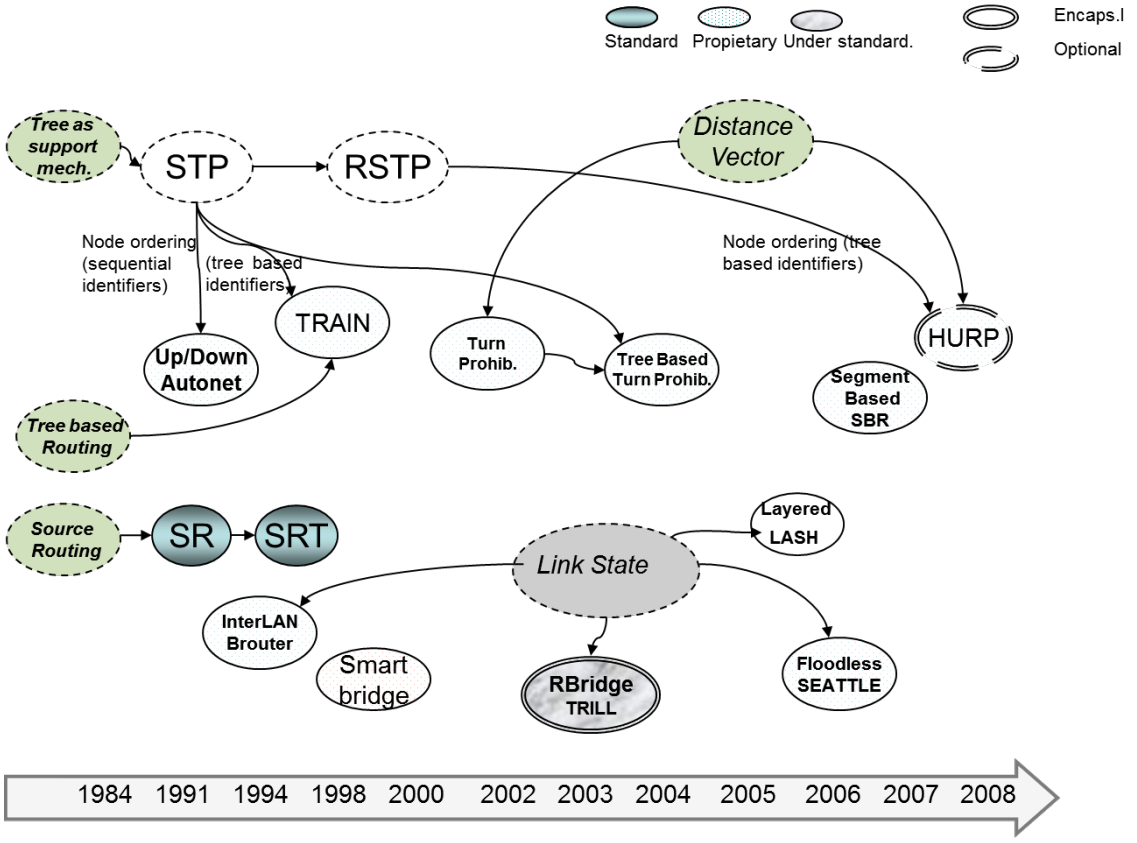


Ilustración 5. Evolución de los puentes enrutadores

2.9.1 Optimal-Suboptimal Routing y BRouter

Optimal-Suboptimal Routing [SC88] es un protocolo de puentes con aprendizaje, propuesto por Sincoskie y Cotton en 1988, de forma que se pueden identificar rutas óptimas o sub-óptimas entre puentes. Los puentes situados en el extremo, puentes hoja (*leaf bridges*) aprenden las direcciones MAC de los hosts conectados y las distribuyen jerárquicamente hacia arriba en el árbol de forma iterativa. BRouter [LG91] es una propuesta de Lin y Gerla que emplea vectores distancia en los bridges para obtener caminos mínimos entre varias LAN, no entre hosts.

2.9.1.1 SmartBridge

SmartBridge [RTA00] es, tras Autonet, el antecedente más claro de los actuales *Routing Bridges* y *Shortest Path Bridges*, pero utilizando protocolos propietarios. Rodeheffer introdujo una arquitectura de puentes que presenta características de enrutador. El encaminamiento se realiza por caminos mínimos y el protocolo entre puentes tiene un conocimiento completo de la topología de la red. Los tiempos de reconfiguración son muy pequeños (10-20 ms.) y permite la utilización de toda la infraestructura. No es compatible con los puentes estándar aunque las tarjetas de interfaz de red son estándar.

2.9.1.2 Link State Over MAC

Link State Over MAC (LSOM) [GDS03] utiliza un protocolo de estado de enlaces para el encaminamiento por caminos mínimos. Los puentes anuncian por difusión su conectividad (anuncio de estado de enlaces) y así obtienen una visión completa de la topología de red. LSOM está orientado a dorsales Metro Ethernet donde un número reducido de puentes intercambia información de encaminamiento de las direcciones MAC de los enrutadores conectados.

2.9.1.3 Routing Bridges (TRILL)

RBridges [RBRIDGE][Per04][PE05][PTY04] está orientado a redes campus auto-configurables formando una subred IP única (para evitar la gestión de direcciones IP y facilitar la movilidad) y se encuentra en estado muy avanzado de estandarización en el grupo de trabajo TRILL del IETF. Los RBridges calculan caminos mínimos entre puentes y soportan encaminamiento por múltiples rutas para repartir la carga. Utiliza enrutamiento IS-IS extendido para capa dos y encapsula las tramas con un encabezado (ver Ilustración 6 y Ilustración 7) que incluye, entre otros campos, un número de saltos. Es completamente compatible con redes 802.1D, mediante la elección de un RBridge designado para el reenvío de tramas de cada LAN, pudiendo formar redes mixtas de puentes estándar y RBridges sin restricciones. Esto se logra incrementando la complejidad y alejándose de la trama estándar Ethernet, mediante un encaminamiento explícito salto a salto indicando cada RBridge en la cabecera de la trama la dirección del RBridge del siguiente salto. El diseño soporta VLANs aunque la compatibilidad con los estándares IEEE 802.1Q es cuestionada en el seno del IEEE 802.1.

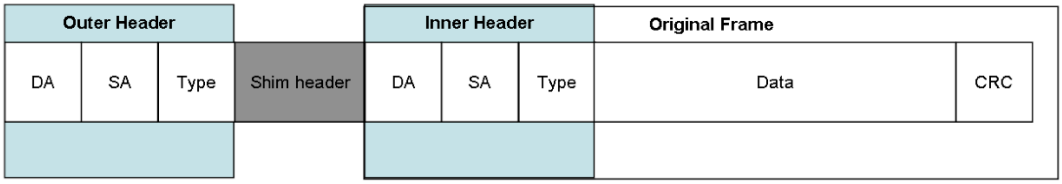


Ilustración 6. Encapsulado de trama en TRILL RBridges [Rut09]

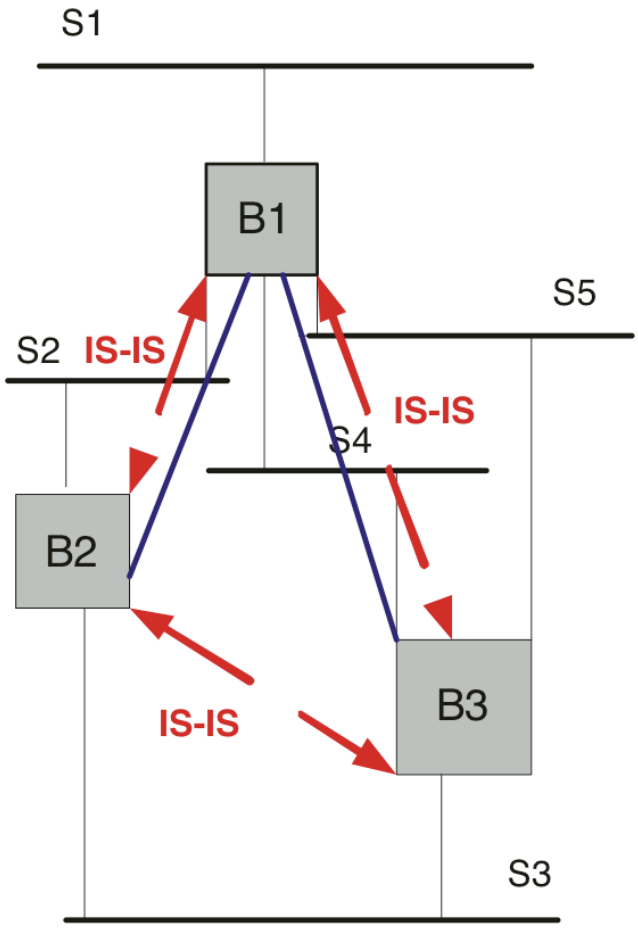


Ilustración 7. Encaminamiento entre RBridges con IS-IS y LANs asociadas [Rut09]

2.9.1.4 IEEE Shortest Path Bridging

IEEE 802.1aq *Shortest Path Bridging* (SPB) [8021aq], el proyecto SPB deriva de MSTP y se basa en optimizar los caminos y la utilización de la infraestructura reemplazando el plano de control MSTP con uno SPB. SPB utiliza instancias múltiples de árbol de expansión enraizadas respectivamente en cada uno de los puentes frontera para obtener de esta forma caminos mínimos entre los puentes. La propuesta es compatible con MSTP y VLANs (802.1Q). Comenzado en marzo de 2005 y estandarizado en marzo de 2012, bajo la denominación *Shortest Path Bridging* se entiende por *shortest path bridging region* una región (en el sentido que tiene en el protocolo MSTP) en la que se establecen caminos mínimos entre los puentes (*shortest path bridges)* que pertenecen a ella mediante el establecimiento de árboles de expansión múltiples unidireccionales, con raíz en cada bridge del área. Para que el aprendizaje de las direcciones MAC funcione correctamente, es necesario :

- Bien que el tráfico se difunda confinado por un único árbol, caso del árbol de expansión,
- Bien que el tráfico de una estación *a* unida a *A* se confine por un árbol enraizado en A y el de una estación *b* se confine por un árbol enraizado en el bridge *B.* Es necesario que los árboles en direcciones opuestas entre dos bridges coincidan: el camino de *A* hasta *B* debe ser idéntico al camino de *B* hasta *A* por el árbol enraizado en *B*. De esta forma las direcciones aprendidas por un puerto pueden utilizarse como destino por el mismo puerto. Suponiendo una VLAN asociada a cada uno de dichos árboles, deben utilizarse las VLAN de forma bidireccional, y deben hacerse sinónimas en lo que se denomina *pairwise shared learning VLANs.*

Se plantea el proyecto como una enmienda con la denominación *Shortest Path Bridging* al estándar 802.1Q. (VLANs). El objetivo del futuro estándar es, textualmente: *"This standard specifies shortest path bridging of unicast and multicast frames, including protocols to calculate multiple active topologies that can share learnt station location information, and support of a VLAN by multiple, per topology, VLAN identifiers (VIDs)"*. Se implementará mediante extensión de las especificaciones existentes de *bridging* para implementar el encaminamiento por caminos mínimos dentro de regiones definidas administrativamente, conservando las funcionalidades actuales. El objetivo es mejorar la utilización de la infraestructura y reducir la latencia. Se considera factible el actualizar mediante software muchos bridges de VLAN actuales. Existe fuerte apoyo de los fabricantes y gran interés de los usuarios en alternativas al protocolo de árbol de expansión RSTP. Los costes de los puentes se considera que no se verán afectados por la nueva funcionalidad y pueden ahorrarse encaminadores IP en muchos casos, reduciendo los costes de las redes. Desde el punto de vista de compatibilidad, debe mantenerse la compatibilidad de la enmienda a la 802.1Q que describimos aquí con los estándares IEEE 802.1D y anterior versión de 802.1Q. Como requisito destacable, los sistemas finales no deben requerir ninguna actualización software o hardware. Para terminar, el proyecto se considera factible tanto técnicamente, basados en la experiencia de tecnologías VLAN, como económicamente.

2.9.1.5 SEATTLE-SEIZE

Una de las propuestas recientes de más impacto es SEATTLE [KCR08], anteriormente conocida como SEIZE (*Scalable Efficient Zero-configuration Enterprise Architecture*) con objetivos similares a TRILL en cuanto a obtener una subred única IP autoconfigurable evitando la gestión de direcciones IP, necesaria si existen subredes aun empleando DHCP. Se centra en evitar totalmente la difusión en capa dos (ARP y replicación de tramas unicast desconocidas) para aumentar la escalabilidad. Desde el punto de vista de encaminamiento no tiene novedad al utilizar encaminamiento de estado de enlaces en capa dos. Para evitar ARP y replicación de tramas, los sistemas finales son registrados automáticamente por los puentes frontera según el hash de su dirección MAC y los guardan en una DHT distribuida entre todos los puentes.

2.10 Puentes para Ethernet Orientada a Conexión (Connection Oriented Ethernet)

El éxito de Ethernet, por su economía y prestaciones, en las redes de transporte ha producido una serie de propuestas que intentan exportar a Ethernet el modelo de las redes SDH y otras redes de transporte de datos basado en conexiones semipermanentes de datos gestionadas por el operador mediante sistemas de gestión. La identificación de las conexiones hace uso intensivo y extiende mucho el tamaño y significados de las etiquetas 802.1Q (VLAN) para permitir la separación de flujos. Describimos a continuación las más destacables. La tabla 2 muestra un resumen comparativo de las principales propuestas.

2.10.1 Global Open Ethernet

Global Open Ethernet (GOE) [Iwa+04] es un proyecto de NEC para redes privadas virtuales de capa dos, reivindicado como autoconfigurable, que se propone como alternativa al encapsulado anidado denominado Q-in-Q (anidamiento sucesivo de etiquetas 802.1Q de VLAN) y a Ethernet sobre MPLS (EoMPLS) por ser ambos sistemas complejos de gestionar. Q-in-Q es además poco eficiente en el uso de la red y los equipos EoMPLS son muy costosos. GOE utiliza conmutación de etiquetas (*tag switching*) para encaminar por la red del proveedor. Se asigna una VLAN ID a cada bridge, para de esta forma utilizar la etiqueta de VLAN asignada (VLAN tag) como dirección de encaminamiento (lo mismo que hace MSTP con las tramas por los árboles de expansión), lo cual en principio es mas parecido a IP que lo que hace MPLS.

En GOE se utiliza, como se muestra en la Ilustración 8 y Ilustración 9, una etiqueta jerárquica de longitud variable (compatible con la de VLAN). Esta etiqueta GOE tiene una parte obligatoria y otras dos opcionales (Customer ID y Vendor ID), desacoplando los clientes y la respectiva VLAN de usuario. La parte obligatoria de la etiqueta se utiliza como dirección de nodo destino para el encaminamiento en los árboles, con la misma longitud que una identidad de VLAN. Como la etiqueta de VLAN estándar es de 12 bit, se produce una compresión del espacio de direcciones de 48(MAC) +12 (VLAN) bits a 12 bits. Esto simplifica y acelera el procesado de tramas en los bridges. GOE es compatible con el protocolo MSTP estándar. Dado que la dirección del conmutador destino se codifica en 12 bit en la etiqueta obligatoria VLAN, en ello se basa la compatibilidad con MSTP. La jerarquía es recursiva y se permite apilamiento sucesivo de las VLAN y de los árboles RSTP/MSTP.

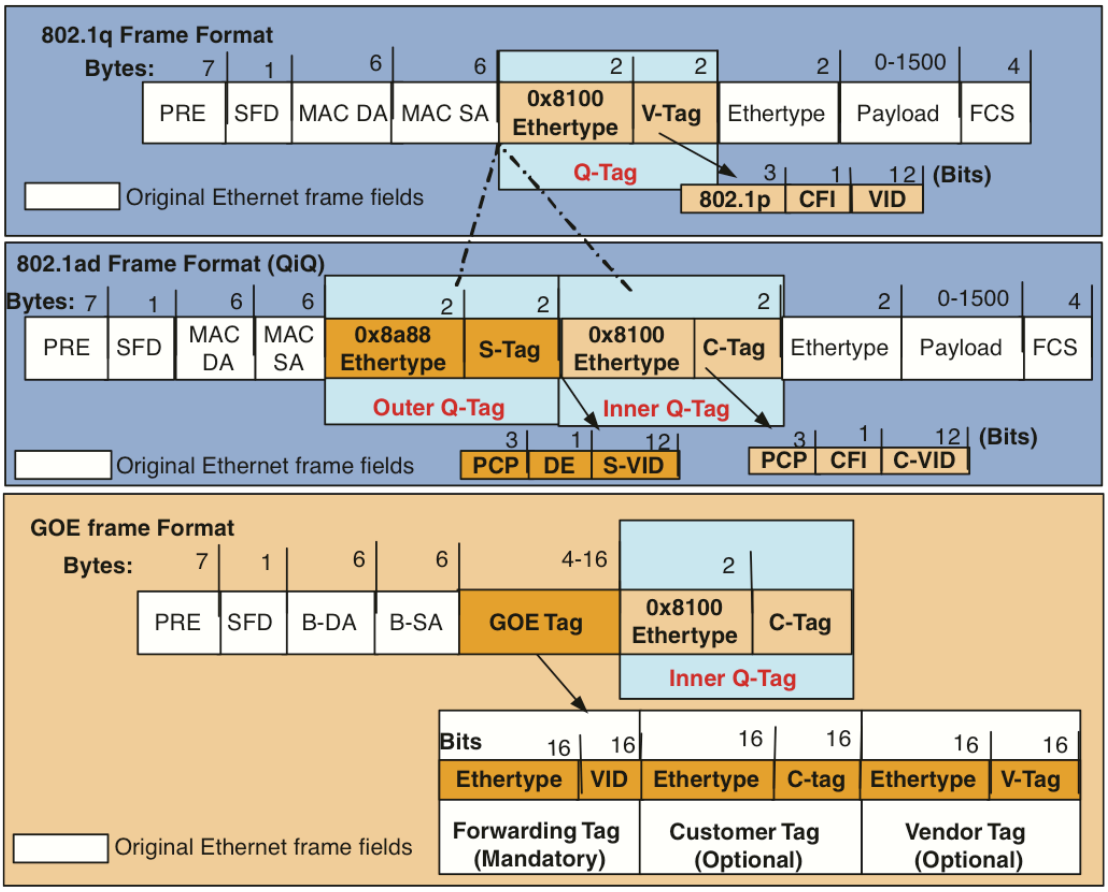


Ilustración 8. Etiquetado jerárquico de tramas en Global Open Ethernet (GOE), Q-in-Q y 802.1Q [Rut09]

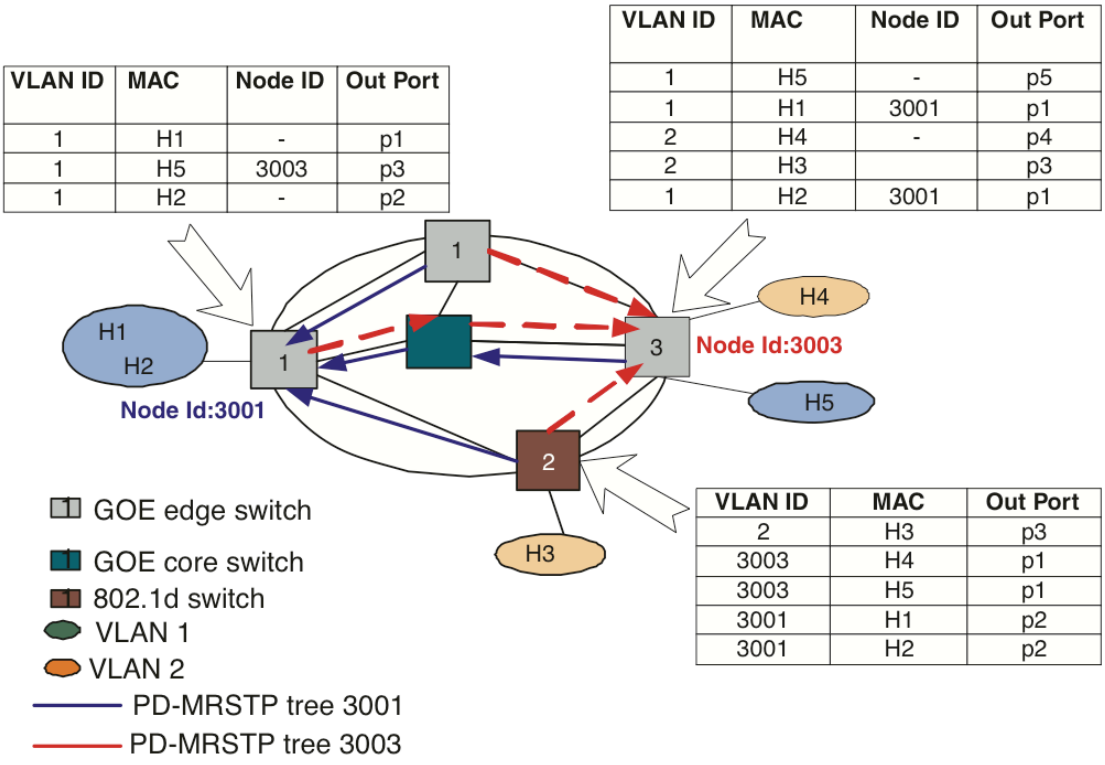


Ilustración 9. Reenvío de tramas en GOE [Rut09]

2.10.2 VLANs automáticas calculadas (*Viking*)

La complejidad de configuración que por un lado presenta MSTP y las posibilidades de utilización de infraestructura que brindan los árboles múltiples de expansión son las empleadas por Viking [Sha+04], una propuesta que realiza cálculo y configuración centralizados de las VLAN y de los árboles MSTP en los puentes mediante SNMP.

Viking es una arquitectura orientada a redes metropolitanas y redes de tipo clúster como las redes de almacenamiento *(Storage Area Networks SAN).* Utiliza árboles múltiples de expansión para lograr la utilización eficiente de la infraestructura. Precalcula varias instancias de árbol de expansión y asigna VLANs a los hosts para optimizar el rendimiento total usando caminos mínimos y enlaces múltiples y redundantes para protección en caso de fallo de enlace. Las rutas entre los nodos se consolidan mediante un algoritmo de agregación de caminos. El elemento director es el *Viking Manager* que configura a los puentes y los sistemas finales. El *Viking Manager* precalcula las rutas entre sistemas finales incluyendo rutas alternativas de reserva. Para optimizar la utilización de la red, las rutas se agregan asignándolas a una VLAN determinada. La configuración de las VLANs y el árbol de expansión correspondiente en los bridges de la red las realiza el *Viking Manager* mediante comandos SNMP a los puentes. Viking requiere modificar el comportamiento de los sistemas finales actuales, dado que los sistemas finales incorporan una lógica de selección de VLAN en la pila de protocolos de red. Los conmutadores Ethernet pueden ser estándar. La supervisión de la red se hace mediante *traps* SNMP que disparan notificaciones al Viking Manager.

2.10.3 Provider Bridges (PB)

Provider Bridges (IEEE 802.1ad) [8021ad], también conocido como Q-in-Q, es una enmienda al estándar IEEE 802.1Q (MSTP) que permite a un proveedor de servicios prestar el equivalente a un servicio de LANs separadas, redes de puente locales o redes LAN virtuales puenteadas, separando a los clientes totalmente. Para ello se distinguen y manejan separadamente e independientemente las VLANs del Cliente (asignadas por el cliente) y las VLAN de Servicio VLANs (asignadas por el proveedor) , para separar el tráfico en las redes respectivas de cliente y del proveedor.

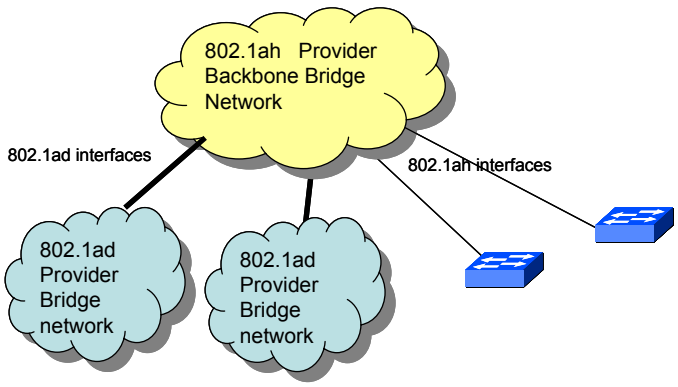


Ilustración 10. Provider Bridges and Provider Backbone Bridges networks

2.10.4 Provider Backbone Bridges y otros

Provider Backbone Bridges (PBB) tiene como objetivo interconectar los *Provider Bridges* (PB) mencionados mediante intercambio de tramas que encapsulan las direcciones, etiquetas VLAN y datos del cliente en un número de instancias del servicio independientes. La Ilustración 11 compara los etiquetados utilizados.

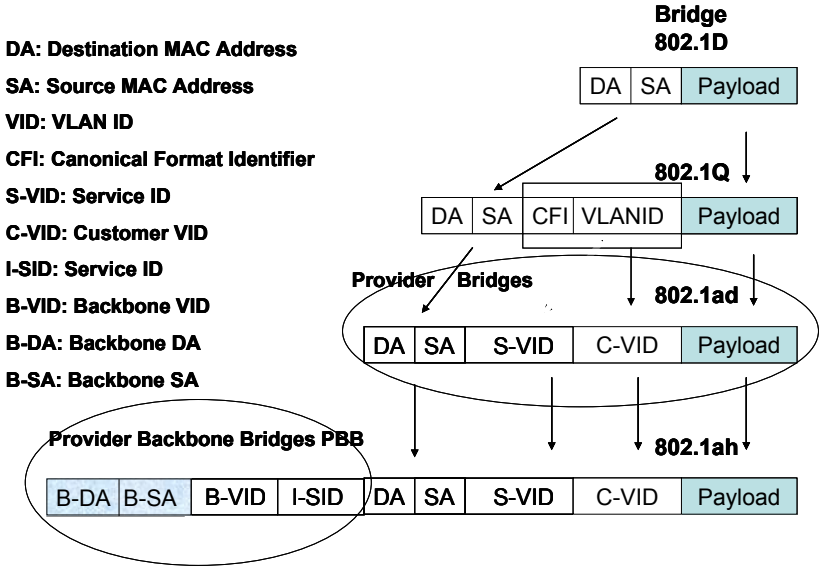


Ilustración 11. Encapsulados IEEE 802.1Q, 802.1ad, 802.1ah

En el proceso de estandarización que ha culminado en la estandarización de SPB se han desarrollado, dentro y fuera del IEEE 802.1aq, diversas propuestas intermedias que se describen brevemente a continuación:

Provider Link State Bridging (PLSB) es un protocolo intermedio en el grupo 802.1ah del IEEE finalmente englobado en SPB. Ofrece reenvío por caminos mínimos (inspirado en 802.1aq) mediante protocolo de estado de enlaces para redes *Ethernet Carrier Grade*, con convergencia rápida y fiable. Su objetivo es reducir el esfuerzo de provisionamiento respecto a PBB, evitar bucles de encaminamiento sin bloquear puertos y soportar tráfico unicast y multicast en capa dos en el contexto de encapsulado MAC-en-MAC de *Provider Backbone Bridges* (PBB).

Provider Backbone Transport (PBT), denominado oficialmente *Provider Backbone Bridges Traffic Engineering* (PBB-TE), inicialmente promovido por Nortel en el IEEE, es un método para ofrecer, provisionado mediante gestión, ingeniería de tráfico de caminos punto a punto Ethernet en redes PBB. Incluye mecanismos de protección de camino, operación y mantenimiento y calidad de servicio (QoS). PBT avanza en la dirección de separar el plano de control de Ethernet de los puentes PB y PBB, actualmente el protocolo MSTP 802.1Q, del plano de datos, introduciendo el control del plano de datos mediante gestión de las conexiones sobre PB y PBB, configurando caminos como en un dispositivo cros-conector.

Provider VLAN Transport (PVT) [KS06] es una propuesta de Huawei-Siemens para Ethernet orientada a conexión. Se basa en el concepto de cros-conexión de VLAN (VLAN-XC), en el que, aparte del reenvío estándar VLAN, una dupla formado por VLAN e identificador destino identifica la cros-conexión, configurable mediante el sistema de gestión de red.

Tabla 2. Comparativa de *Connection Oriented Ethernet* [Rut09]

Solution	Tunneling/Service Forwarding	Technology	QoS	Protection	Traffic aggregation	OAM	Supported connections			Standardisation status
							1:1	1:N	N:N	
Ethernet (VLAN bridging)	MAC+VLAN	Connectionless; address resolution based on MAC learning/broadcasts; STPapproaches for loop avoidance; VLAN configuration performed manually (no integrated control/management plane)	802.1p	1:1	16 M VLANs per provider network	802.1ag, manual	**Yes**	**Yes**	**Yes**	Standardised
PBB-TE	Tunnel: MAC+B-VID; Service: I-SID	Connection-oriented; supports both raw Ethernet or 802.1ad; address resolution only required on the edges (no customer MAC learning in the core); requires sophisticated management plane for tunnel creation and bandwidth management	PCP/DE	1:1; 802.1ag monitoring; ITU-T G.8031 – SG 15 Ethernet protection	16M VLAN per tunnel (60-bit tunnel identifier)	802.1ag, ITU-T Y.1731 - SG 13 Ethernet OAM, manual	Yes	Yes	Yes	PBB as draft (802.1ah); PBB_TE under discussion
VXC	Ingress Port+VLAN	Connection-oriented, similar to ATM switching; re-engineering of QiQ; MAC learning only at the edges (no customer MAC learning in the core); requires sophisticated management plane for tunnel creation and bandwidth management on a hop-by-hop basis	802.1p	1:1	16M VLAN per port	Manual	Yes	No	No	Early discussion, IEEE, IETF, ITU-T
T-MPLS	Tunnel: MPLS LSPs; Service: PW/VPLS label at bottom of stack	Connection-oriented; subset of MPLS (no IP functionality); specifically designed for packet-based service support by means of point-to-point connections; requires sophisticated management plane for tunnel creation and bandwidth management	CS tags	1:1; possibly fast rerouting; ITU-T recommended Y.1720/G.813 linear protection switching	20-bit label	ITu-T Y.1711 (ITU-T OAM mechanisms for MPLS)	**Yes**	**Yes**	**No**	ITU-T Work in progress (SG 15)

2.11 Protocolos de prohibición de giros

Los protocolos basados en prohibición de giros surgen como alternativa más eficiente a los protocolos de árbol de expansión inhabilitando en menor medida los enlaces al prohibirse determinados *giros* en la red, en lugar de inhabilitar completamente los enlaces redundantes. Como se indica en la Ilustración 12, aparecen por primera vez en 1991 con Autonet (AN-1) [Sch+91]. Los protocolos de prohibición de giros, cuya esencia es la elección de un conjunto de giros en la topología que impida los bucles, obtienen ventaja si tienen un conocimiento completo de la topología. Por ello predominan los protocolos centralizados frente a los distribuidos y normalmente los superan en prestaciones resultando en un menor porcentaje de giros prohibidos. Los protocolos de prohibición de giros distribuidos como *Up/Down*, se apoyan frecuentemente en la construcción previa de un árbol de expansión para el cálculo de giros prohibidos. Los protocolos que se describen son propietarios y no son compatibles con los puentes estándar 802.1D.

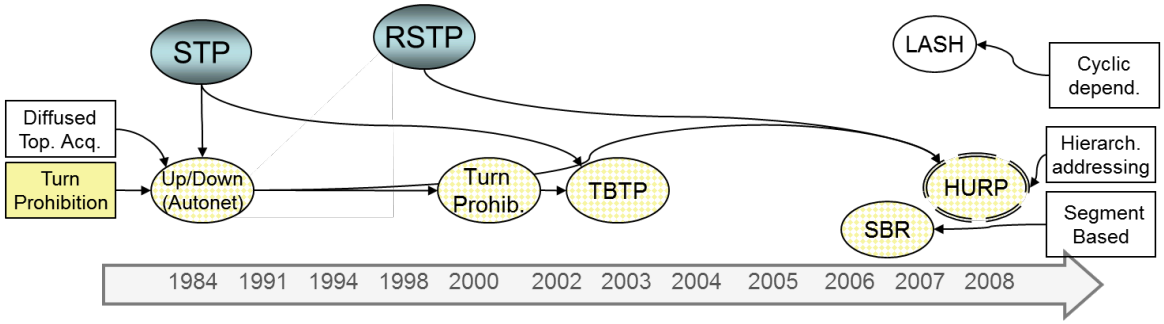


Ilustración 12. Evolución de los protocolos de prohibición de giros

Los protocolos de Prohibición de Giros operan normalmente en dos fases: en la primera se define el conjunto de giros prohibidos y posteriormente se construyen las tablas de encaminamiento. La definición de los giros prohibidos, en los protocolos que utilizan un árbol de expansión auxiliar, consta a su vez de tres fases: construcción del árbol de expansión, etiquetado de nodos según el árbol de expansión y algoritmo de definición del conjunto de giros prohibidos.

2.11.1 Encaminamiento arriba/abajo (Up/Down routing)

Autonet emplea por primera vez el protocolo *Up/Down.* Este protocolo utiliza un esquema simple y distribuido de prohibición de giros. Es un algoritmo libre de bucles que se basa en asignar un sentido a cada enlace de la topología y prohibir los giros abajo-arriba. Up*/Down* puede producir congestión en algunos puntos de la red cercanos al bridge raíz y dado que se basa en árbol, no necesariamente obtiene encaminamiento por caminos mínimos.

El encaminamiento arriba/abajo (*Up/Down routing*) está basado en asignar una dirección a todos los enlaces de la red dependiendo de la posición del extremo del enlace (nodo) en el árbol de expansión auxiliar construido para asignar identificadores a los nodos (hacia arriba si el extremo está mas cercano al puente raíz y hacia abajo si es al contrario).

Mediante un protocolo de árbol de expansión se asignan identificadores crecientes a los puentes partiendo del puente raíz y descendiendo nivel a nivel. Los enlaces entre nodos a la misma altura reciben la orientación según la identidad del puente sea mayor o menor. El encaminamiento arriba/abajo evita los bucles porque un bucle siempre contiene dos giros abajo/arriba y

arriba/abajo. Por lo tanto, si se obliga a que en cualquier recorrido, una vez que se ha girado hacia abajo, no pueda volver a girar hacia arriba, se evitarán los bucles en las tramas enviadas. Una ruta legal es la que nunca usa/atraviesa un enlace en la dirección hacia arriba después de haber usado uno hacia abajo. Es decir, prohibiendo los giros abajo-arriba se evitan los bucles. La efectividad del encaminamiento arriba/abajo depende principalmente de la elección del árbol de distribución, y en particular del puente raíz. Para la topología típica de las redes campus estructuradas jerárquicamente esto tiene menor incidencia que para una red mallada arbitraria. El problema del encaminamiento arriba/abajo es que no garantiza rutas óptimas, y que se hace menos eficiente a medida que la red se hace más compleja. Silla y Duato [SD97] utilizaron una variante de encaminamiento arriba/abajo en redes de estaciones de trabajo (Networks of Workstations (NOW)). Esta variante, de mejor rendimiento, se basa en la utilización de canales adicionales (virtuales) a los enlaces normales para aumentar las posibilidades de elección de rutas mínimas y evitar de esta manera bloqueos, pero subsisten los inconvenientes básicos del encaminamiento arriba/abajo convencional. El encaminamiento arriba/abajo mejora la efectividad del Árbol de Expansión, pero la congestión en las cercanías del puente raíz permanece.

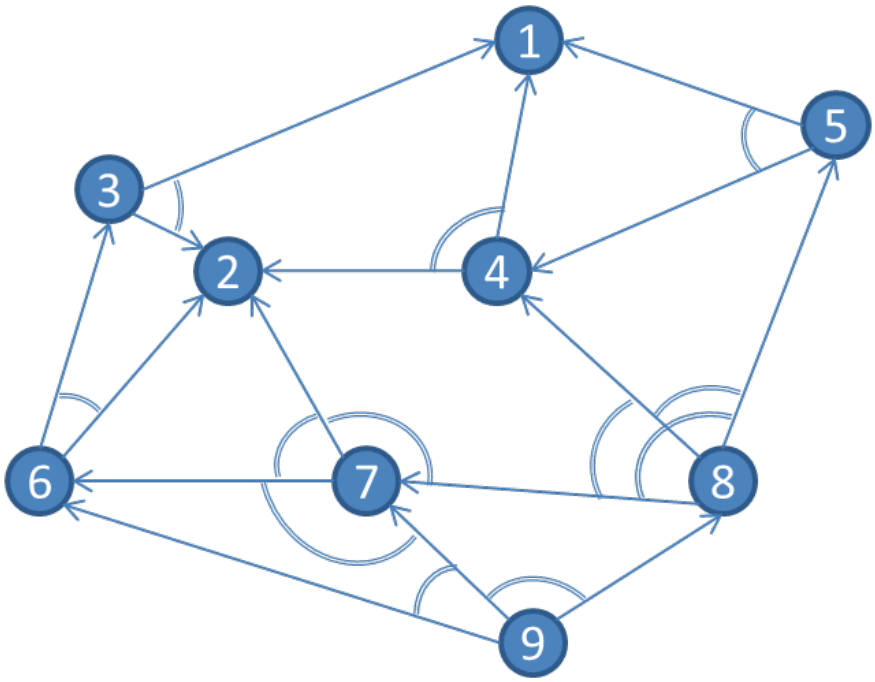


Ilustración 13. Giros prohibidos en el encaminamiento Arriba/Abajo

2.11.2 Otros algoritmos de Prohibición de Giros

2.11.2.1 Turn Prohibition

Aunque utilizamos el término de *prohibición de giros* para todos los protocolos de esta categoría, el algoritmo denominado *Turn Prohibition (TP)* aparece en 2002 como una evolución del encaminamiento Up/Down para mejorar sus prestaciones al precio de centralización completa y mayor complejidad [SKZ02].

Este algoritmo actúa iterativamente sobre los nodos de la topología, seleccionando primero los nodos de menor grado para prohibir todos los giros del nodo. Prohibiendo los giros en los nodos de

menor grado se evitan bucles de forma más efectiva que en nodos de mayor grado, dado que el número de giros posibles en un nodo crece con el cuadrado del grado. A fin de evitar que la red se desconecte se propone un algoritmo mejorado que evita prohibir los giros que puedan resultar en una red inconexa. El rendimiento de TP es alto, suprimiendo como máximo 1/3 del total de giros posibles de la red y normalmente muchos menos, dependiendo de la topología. Existe una versión ponderada del algoritmo, que pondera los enlaces con pesos relativos, resultando en la prohibición de menos de la mitad de los enlaces en media ponderada. La complejidad computacional del protocolo es $O(N^2d)$ siendo d el grado máximo de los nodos de la red, lo que limita su escalabilidad. Utiliza encaminamiento basado en el algoritmo de Bellman-Ford empleando tablas indexadas con el puerto de entrada, a fin de aplicar las restricciones de giros a determinados puertos de salida en función del puerto de entrada. Así pues, para los paquetes que llegan por un puerto de entrada puede existir distinta ruta de salida al mismo destino que para los recibidos por otro puerto de entrada.

2.11.2.2 Tree-based turn prohibition

Tree-based Turn Prohibition (TBTP) [Pel+04], es una evolución de TP. TBTP se apoya en un árbol de expansión para la convergencia del algoritmo y como referencia para prohibir o permitir giros según un giro entre en el árbol o abandone el árbol. Prohíbe como máximo la mitad de los giros posibles, mejorando al aumentar el grado medio de los nodos. La mejora que obtiene es proporcional al grado del nodo. TBTP tiene una variante que es compatible y miscible con puentes 802.1D. Existe una versión distribuida de TBTP (dTBTP) [Pel+06], algo menos eficiente, pero de complejidad computacional tan alta como la versión centralizada.

2.11.2.3 Segment-based routing

Segment-based Routing (SBR) [Mej+06] es una propuesta para redes de interconexión de altas prestaciones de tipo malla y toroide. Es un encaminamiento determinista que divide la topología en subredes y las subredes en segmentos, colocando restricciones de giros bidireccionales dentro de un segmento, sin necesidad de canales virtuales (ver sección siguiente). La mejora sobre *Up/Down* es de 1,8 veces. Se colocan restricciones de giro localmente en cada segmento, se gana así flexibilidad en la asignación de restricciones de giro en cada segmento, independientemente de los demás segmentos de la red.

La aplicación de restricciones de giros dentro de un encaminamiento de caminos mínimos mediante el algoritmo de Djikstra no es posible porque se reducen el número de rutas de camino mínimo posibles en las topologías. El impacto en la longitud media de los caminos con restricciones de giros será perceptible o no según el número de rutas alternativas existentes, el cual depende del grado medio de los nodos. En [FE04] se estudia un reciente método para encaminar aplicando restricciones de prohibición de giros en el que la complejidad crece desde $O(N^2)$ a $O(E^2)$, siendo N y E respectivamente el número de nodos y de enlaces.

2.11.2.4 Grafos de dependencia cíclica

Un método para evitar de bucles muy común en las redes de interconexión con *wormhole routing* utiliza grafos de dependencia cíclica [DS87] para establecer un orden de precedencia entre canales. Se cambia la trama de un *canal* a otro cuando puede producirse un ciclo. Para evitar el posible retorno posterior al mismo canal que produciría un ciclo, el encaminamiento se restringe debiendo visitarse los canales en orden creciente o decreciente para así eliminar los ciclos en el grafo de dependencia cíclica. Se ha utilizado en multicomputadores [DS86] entre otros campos. [Dua91] propone el uso de canales virtuales adicionales para evitar las limitaciones del procedimiento de

Dally y reducir la longitud de los caminos, que deja de ser mínima al aplicar las restricciones anti bucle.

Sancho [San+02] propone un método libre de bucles para redes *Infiniband* utilizando pistas virtuales y niveles de servicio, basado en *Up/Down*. Es interesante la diferencia con *Up/Down*: los giros prohibidos (de tipo abajo/arriba) detectados (los que presentan una dependencia cíclica en el grafo) se marcan y posteriormente se obvian realizando un cambio a una pista virtual distinta (en orden siempre creciente o decreciente). Un paquete que circule por la pista virtual 0 continúa hasta llegar a un giro prohibido. Entonces lo realiza y se cambia a la pista virtual 1. El camino con mayor número de cambios de pista es el que determina el numero de pistas mínimo a implementar. Los cálculos muestran que en redes de hasta 256 nodos son suficientes tres pistas.

Layered Shortest Path Routing (LASH) [SL02] es un algoritmo de caminos mínimos para topologías regulares e irregulares. LASH está inicialmente orientado a redes de interconexión pero es también aplicable in redes irregulares Ethernet. Los conjuntos de direcciones (origen, destino) se agrupan en capas virtuales libres de bucles. La prevención de bucles se realiza mediante capas virtuales. Si el numero de enlaces es *N-1* basta con una sola capa virtual de interconexión. El máximo número de capas necesario es *B/2* siendo *B* el número de puentes.

3 Encaminamiento mediante prohibición de giros y direccionamiento jerárquico

En este capítulo se describen las contribuciones realizadas en protocolos que evitan los bucles mediante la prohibición de giros. Se describe el protocolo HURP y los trabajos complementarios que llevan a dicho protocolo, desarrollados en colaboración con mi director de tesis. Tras la introducción del tema se describen, primero, los trabajos previos y complementarios al protocolo que desembocan en el mismo, a continuación, el protocolo HURP y, finalmente, los resultados de las evaluaciones.

3.1 Introducción

El problema principal a resolver en los conmutadores Ethernet avanzados es poder enviar tramas por toda la red sin que se produzcan bucles. La difusión en todos los puertos de cada conmutador de las tramas *broadcast* dificulta enormemente este objetivo, precisándose mecanismos de prevención de bucles.

Como se indica en la introducción de la presente Tesis, en 2005 existían dos propuestas principales para implementar los conmutadores de camino mínimo o *routing-bridges*: realizar encaminamiento *Shortest Path* entre puentes mediante protocolos de estado de enlaces o implementar arboles múltiples de expansión enraizados en cada nodo utilizando una VLAN por puente asociada al mismo. Una tercera vía, en realidad predecesora, era el uso de protocolos de prohibición de giros como *Up/Down*, definido inicialmente en Autonet [Sch+91] y que posteriormente se generaliza en la categoría de protocolos de prohibición de giros como [SKZ02] [Pel+04].

3.2 Protocolos de prohibición de giros

Frente a los protocolos de árbol de expansión que bloquean determinados enlaces de la red, los protocolos de prohibición de giros evitan la generación de bucles seleccionando un conjunto de *giros* de entre todos los posibles de la topología de forma que se impidan los bucles de tramas. Un giro no es más que la concatenación de un enlace de entrada y un enlace de salida concretos en un nodo. De manera que cuando se habla de un giro prohibido, en realidad lo que se pretende indicar es que el nodo en cuestión no puede reenviar un paquete recibido por dicho enlace de entrada a través del enlace de salida que completa ese giro prohibido. Ellos no implica que dichos enlaces, tanto el de entrada como el de salida no pueden utilizarse para encaminar otras tramas, siempre y cuando no coincidan juntos. De esta manera, se evita el bloqueo completo de enlaces y aumenta, en consecuencia, el rendimiento. Es importante puntualizar que se produce un bucle de una trama cuando ésta pasa dos veces por el mismo enlace, no cuando ésta pasa dos veces por el mismo nodo.

El elemento clave que permite distinguir los distintos protocolos basados en prohibición de giros reside en el mecanismo de selección del conjunto de giros prohibidos. Cuanto menor sea el número de giros prohibidos totales, mayor capacidad de selección de caminos quedan abiertos al algoritmo de enrutamiento y, por tanto, mayor será el rendimiento final que se alcance. De entre ellos, los más destacados (comentados en el capítulo de estado del arte) son el protocolo *Up/Down* (*Autonet*), el protocolo TP (*Turn Prohibition*) y el protocolo TBTP (*Tree-based Turn Prohibition*).

Tanto con TP como con TBTP se ha demostrado que sus algoritmos de selección de giros prohibidos permiten garantizar topologías de difusión libres de bucles bloqueando menos de un tercio del total de giros posibles de la topología frente a valores típicos de entre el 70% y el 90% en STP. Sin embargo, se trata de algoritmos centralizados y complejos que presentan claros problemas tanto de escalabilidad como de reconfiguración frente a fallos en la red. El mecanismo original, *Up/Down*, no produce tan buenos resultados como sus sucesores, pero es mucho más sencillo de implementar, sin embargo, requiere modificaciones tanto hardware como software en los sistemas finales lo que choca con uno de los requisitos básicos planteados en este trabajo como es la transparencia de las soluciones aportadas frente a los sistemas finales.

Por tanto, el objetivo de la contribución presentada en este capítulo consiste en diseñar una arquitectura y su correspondiente protocolo de enrutamiento que partiendo de la idea básica de *Up/Down* permita alcanzar rendimientos cercanos a un enrutamiento puro de camino más corto (*shortest path*) sin renunciar a las características de autoconfiguración, transparencia con sistemas finales y, por supuesto, respetando el formato estándar de la trama Ethernet y el aprendizaje transparente de los conmutadores.

El protocolo propuesto, denominado protocolo HURP (*Hierarchical Up/Down Routing Protocol*) consta de tres componentes principales: un mecanismo de direccionamiento jerárquico, un algoritmo de enrutamiento derivado de los algoritmos de vectores de distancia y un protocolo para la asignación de direcciones y la autoconfiguración de la red como un núcleo HURP al que se adosan islas de conmutadores estándar. El resultado es pues una arquitectura auto-configurable, transparente a los sistemas finales y compatible (en modo islas) con puentes o conmutadores 802.1D.

Como mecanismo de direccionamiento jerárquico se propone utilizar una adaptación del direccionamiento estándar MAC. Utilizando la posibilidad de definir direcciones con significado local mediante una marca (*flag*) en el primer octeto de la dirección, se utilizan los 46 bits restantes para definir un esquema de direccionamiento jerárquico que aporte la información necesaria para aplicar el mecanismo de enrutamiento elegido al cálculo de rutas y sirva de base, a su vez, para poder implantar el esquema derivado de *Up/Down* para el cálculo de giros prohibidos que garanticen la difusión sin bucles. También se utilizan para el reenvío directo de tramas (sin difusión ni consultas de tablas) a través del árbol de expansión, por el simple mecanismo de descodificación de las direcciones destino que realiza cada conmutador.

Se propone la utilización de un algoritmo de encaminamiento de tipo vector de distancias extendido para respetar las restricciones de uso de determinadas combinaciones de enlaces (entrada-salida) impuestas por el algoritmo de giros prohibidos.

Finalmente, se propone la extensión del protocolo RSTP (algoritmo de árbol de expansión) como mecanismo para la asignación de las direcciones jerárquicas a los nodos, por tanto, se genera un árbol clásico, pero con direcciones locales que aportan información topológica sobre la situación concreta de cada nodo dentro de la jerarquía lógica definida por el árbol RSTP. Además, RSTP permite extender el núcleo de red formado por los conmutadores HURP con islas de puentes o conmutadores estándar que se unen a dicho núcleo, al que se considera como raíz lógica de su correspondiente árbol de expansión estándar.

En las secciones sucesivas se describen en detalle cada uno de los componentes, los resultados del análisis de rendimiento obtenidos mediante simulaciones y las conclusiones.

3.3 Encaminamiento Up/Down en redes Ethernet

En nuestra investigación partimos del protocolo *Up/Down* buscando su aplicación en redes Ethernet estándar, por lo que se utilizan solamente sus mecanismos básicos. A continuación, se describen los principios básicos de funcionamiento de la prohibición de giros del protocolo *Up/Down*, basada en los identificadores de nodo, y la adaptación realizada para HURP.

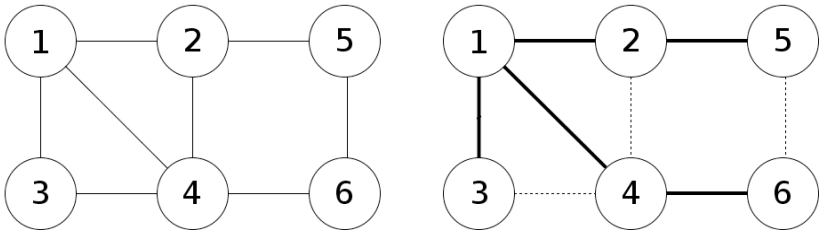


Ilustración 14. Grafo original (izqda), arból de expansión (dcha)

La Ilustración 14 muestra un ejemplo de red sencilla, formada por seis nodos y la topología activa que se deriva de aplicarle el algoritmo de árbol de expansión tomando el nodo etiquetado como "1" como nodo raíz, las líneas discontinuas indican que los enlaces 2-4, 3-4 y 5-6 quedarían bloqueados por STP.

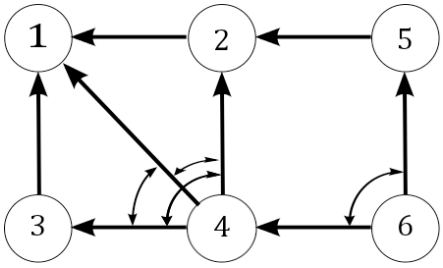


Ilustración 15. Giros prohibidos

Por su parte, en la Ilustración 15 se muestra el principio de operación de *Up/Down*, los arcos indican giros prohibidos en la red y nuevamente se toma el nodo etiquetado como "1" como origen del cálculo; en este caso, quedarían prohibidos los giros 1-4-2, 1-4-3, 2-4-3 y 4-6-5. El nodo etiquetado como "4" resulta ser el más afectado por el algoritmo ya que la mitad de sus giros posibles resultan prohibidos, pero no queda ningún enlace bloqueado. Es importante destacar también como el camino 4-3-1-2-4-6 no contiene bucles; aunque el nodo 4 es visitado dos veces, ningún enlace es atravesado dos veces en la ruta.

3.3.1 Formalización de Up/Down

Consideremos una red modelada como un grafo dirigido *G* compuesto de nodos *{a,b, …, n}* y enlaces bidireccionales, donde la dupla *n1-n2* describe el enlace directo que une los nodos n_1 y n_2.

Definimos el grado de un nodo *d* como el número de enlaces que conectan ese nodo con sus nodos vecinos (por tanto, equivale al número de nodos vecinos).

Un camino $P=\{n_1,n_2,\ldots,n_n\}$ es la secuencia de nodos sucesivamente conectados por enlaces directos, es decir, vecinos dos a dos, que permite avanzar desde el nodo origen "n_1" al nodo destino "n_n". A diferencia de la teoría de grafos, consideramos que existe un ciclo en el camino cuando el primero y último enlace del (sub)camino coinciden, en vez de que el primero y último nodo coincidan. Por ello, un nodo puede ser visitado repetidamente sin crear necesariamente un ciclo (bucle).

Se define un giro como el par de enlaces que conectan un nodo cualquiera con dos de sus nodos vecinos. Se representa mediante la tupla *(a,b,c)* y puede verse como el trayecto que sigue una trama que accede al nodo *b* a través del enlace *a-b* y es reenviada hacia el nodo *c* vía el enlace *b-c*. Salvo indicación en contra, los giros son por defecto simétricos, por lo que el giro *(a,b,c)* es idéntico al giro *(c,b,a)*. El número de giros posible alrededor de un nodo de grado *d* aumenta cuadráticamente con *d* y se expresa como $d \bullet (d-1)/2$.

Supongamos que se construye un árbol de expansión $T(G) = (V_T;E_T)$, donde V_T representa el conjunto de nodos y E_T el conjunto de enlaces del árbol que da conectividad al grafo *G*. Los enlaces pertenecientes al árbol se denominan enlaces de árbol (*tree-links*). Los demás enlaces se denominan enlaces de cruce (*cross-links*).

El objetivo es la obtención de un conjunto de giros prohibidos *ST(G)* que rompa todos los posibles ciclos del grafo y, por tanto, permita difundir tramas sin peligro de bucles.

Up/Down es la aproximación más simple para la construcción de este conjunto de giros *ST(G)* libre de ciclos. Con ayuda del árbol de expansión *T(G)* los nodos son ordenados y se les asigna un identificador según su distancia al nodo raíz (medida en número de saltos) de manera que exista una regla clara que permita definir dados dos identificadores cuál es mayor. Los nodos situados en el mismo nivel respecto al nodo raíz se ordenan arbitrariamente (por ejemplo en función de su identificador de puente). Una vez que los nodos tienen asignado un identificador, un enlace *a-b* se considera hacia arriba (*up*) si b es mayor que a $(b > a)$, en caso contrario, $(a > b)$ el enlace *a-b* se considera hacia abajo (*down*). De esta manera un giro *(a,b,c)* se considera arriba/abajo (*up/down*) si se cumple que $a < b$ and $b > c$. Por el contrario, si $a > b$ and $b < c$ el giro es abajo/arriba (*down/up*).

Para que se produzca un ciclo en el trayecto de una trama por la red, deben existir al menos un giro arriba/abajo (up/down) y otro abajo/arriba (down/up) que complete el ciclo, por ello, prohibiendo todos los giros de uno de los dos tipos, normalmente los del tipo abajo/arriba (down/up), se rompen todos los posibles ciclos en la topología a la vez que se garantiza la alcanzabilidad de todos los nodos.

El protocolo *Up/Down* se basa en la asignación de identificadores a los nodos siguiendo una secuencia desde la raíz, de manera que cuanto más lejos de la raíz se encuentre un nodo más alto será su identificador. En el ejemplo de la Ilustración 15, si aplicamos el criterio de prohibir los giros de tipo abajo/arriba (aquellos en los que el identificador del nodo intermedio es el numeral más alto y, por tanto, indica que se trata del nodo más alejado de la raíz de los tres que forman el giro), resultan prohibidos los giros: (3-4-2), (3-4-1), (2-4-1) y (4-6-5). Cada giro prohibido previene la formación de uno o más bucles en la red, por ejemplo, prohibiendo el giro (2,4,3) se garantiza que ninguna trama realizará el bucle 1-2-3-4 y prohibiendo el giro (4,6,5) se impiden los bucles 2-5-6-4, 1-2-5-6-4 y 1-2-5-6-4-3. Es importante destacar que los identificadores que utiliza U*p/Down* no aportan ningún tipo de información topológica, son *planos*. Además, el criterio de asignación de los mismos puede tener un efecto importante en la cantidad de giros prohibidos que resulte de su aplicación.

Otros protocolos de prohibición de giros como *Turn Prohibition* (TP) y *Tree Based Turn Prohibition* (TBTP) proponen mecanismos para asignar los identificadores mediante algoritmos de proceso sistemático (e iterativo) de la topología. Se van seleccionando los nodos de manera que se minimice el número de giros prohibidos resultante. Por ejemplo, en TP se toma como criterio el grado de los nodos (un nodo de grado alto situado muy abajo en el árbol producirá muchos giros prohibidos). Estos algoritmos garantizan que con un porcentaje de giros prohibidos bajo (hasta un tercio) se consiguen romper todos los posibles bucles en la topología, pero requieren el conocimiento completo de la misma (no son distribuibles), y cálculos complejos para garantizar que el proceso de asignación de identificadores no divide la red, por lo que se plantea partir del principio básico de

Up/Down que simplemente requiere de la formación de un árbol de expansión (algo ya disponible en Ethernet).

Una vez que se dispone de una topología activa libre de bucles, los nodos pueden intercambiar la información de encaminamiento necesaria para calcular las mejores rutas, pero siempre respetando los giros prohibidos.

3.4 El protocolo HURP (Hierarchical Up/Down Routing Protocol)

El protocolo HURP consiste en la aplicación del principio de encaminamiento sin bucles basado en prohibición de giros utilizando un direccionamiento jerárquico compatible con el estándar MAC de Ethernet.

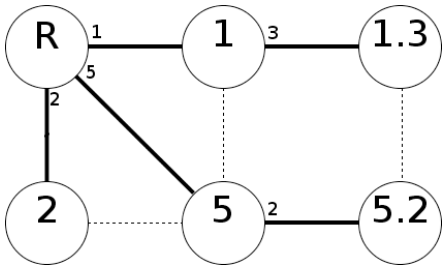


Ilustración 16. Árbol de expansión y direcciones jerárquicas asignadas

La Ilustración 16 representa un posible árbol de expansión y las correspondientes direcciones jerárquicas asignadas a los nodos. En este caso se ha utilizado el número de puerto del enlace hacia la raíz como sistema de numeración. Por ejemplo, el nodo situado abajo a la derecha tiene la dirección 5.2 porque su camino hacia la raíz del árbol utiliza el puerto 2 del nodo 5, que a su vez, recibe ese identificador por utilizar el puerto 5 de la raíz para llegar a ella.

El direccionamiento jerárquico aporta información topológica sobre la localización de los nodos en la red que resulta de utilidad para mejorar el rendimiento global del protocolo. Un giro prohibido aplicando las reglas básicas del protocolo puede resultar útil (en el sentido de que no produzca un bucle en la red) dependiendo de cual sea el destino final de la trama. Por ejemplo, cuando el giro termine en la rama del árbol destino de la trama que se pretende reenviar; dado que el árbol, representa caminos mínimos, una vez que la trama llegue a la rama que contiene el destino ya no la abandonará nunca, por lo que ese giro prohibido no es problemático y puede permitirse para esa trama (aunque no para otras).

Por ejemplo, tomando de nuevo la Ilustración 16 y el nodo etiquetado como '5.2', las tramas cuyo destino sea el nodo '1.3' podrían realizar el giro prohibido por las reglas básicas del protocolo (2,5,1) ya que una vez en el nodo '1' ya no pueden abandonar la rama del árbol y posibilitar así un bucle. De hecho ocurre lo mismo con el giro (5,5.2,1.3) por las mismas razones indicadas.

Además, al usar identificadores jerárquicos, los encaminadores pueden realizar agregación de nodos y 'anunciar' un único identificador en el proceso de intercambio de información previo al cálculo de rutas reduciendo de esta manera la sobrecarga del proceso.

Por último, con identificadores jerárquicos, se podría realizar el proceso de reenvío, sin necesidad de recurrir a la difusión ni al aprendizaje de direcciones MAC, utilizando el árbol como mecanismo de último recurso, simplemente descodificando la dirección jerárquica y reenviando en consecuencia hacia arriba por el árbol hasta alcanzar la rama común al destino y de vuelta hacia abajo por la rama que contiene al destino.

3.4.1 Direccionamiento jerárquico HLMAC

Al igual que *Up/Down*, HURP asigna identificadores a los nodos en función de su distancia a la raíz del árbol seleccionado, sin embargo, los identificadores elegidos comportan información topológica jerárquica. El sistema de direccionamiento HLMAC es aplicable tanto a equipos de usuario (sistemas finales) como a conmutadores y puentes aunque en HURP solo se aplica a estos últimos para garantizar la transparencia con respecto a los usuarios. Veremos más adelante, que esto no afecta a la posible aparición de bucles en la red.

En esta sección se describe el mecanismo de asignación de direcciones, basado en una extensión del protocolo RSTP, que de hecho, es el responsable de construir y mantener el árbol de expansión básico de Ethernet en el que nos apoyaremos para aplicar la prohibición de giros.

3.4.1.1 Direcciones HLMAC

Las direcciones HLMAC se definen como un subconjunto del espacio de direccionamiento MAC clásico, de manera que se garantiza su coexistencia con equipos antiguos ajenos a este nuevo esquema ya que se respeta tanto el tamaño como el formato de las mismas.

Se trata de direcciones de 48 bits (6 octetos), indistinguibles de cualquier dirección Ethernet estándar salvo por llevar el segundo bit (o bit 1) del primer octeto de la dirección puesto a '1' (el *flag* o marca de direccionamiento local).

La Ilustración 17 muestra el formato de una dirección MAC estándar y la configuración de los dos primeros bits del primer octeto, el bit 0 puesto a '0' para indicar que se trata de una dirección individual (sería de grupo o *multicast* si estuviera a '1') y el bit local para indicar que el resto de la dirección debe interpretarse según un esquema diferente.

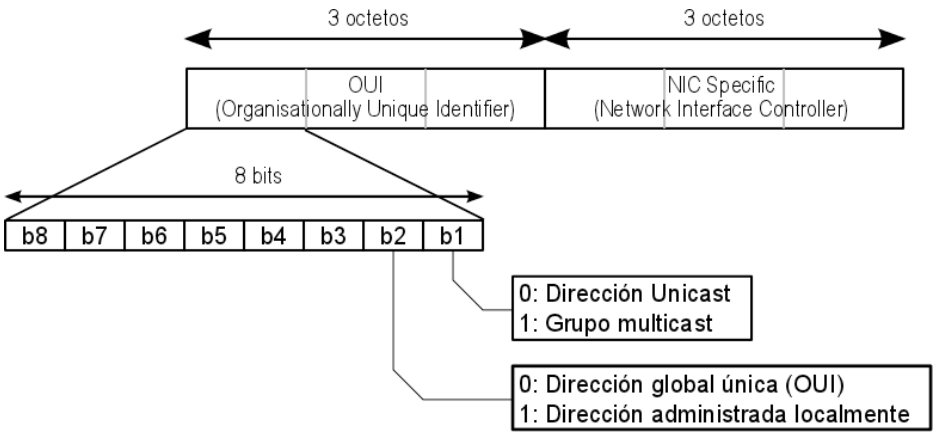


Ilustración 17. Esquema de direccionamiento MAC 802.1D

El resto de la dirección, los 46 bits restantes, se estructuran en seis niveles jerárquicos, uno por octeto, de manera que el primer nivel solo dispone de 6 bits (para alojar hasta 64 equipos), mientras que cada uno de los demás puede alojar hasta 256. Cada nivel se corresponde con un salto de distancia hasta la raíz a la que se le asigna como dirección la primera dirección disponible (todo '0's) para no desperdiciar un nivel completo con una sola máquina. De esta manera podemos construir árboles de hasta seis niveles de profundidad. De este modo, asumiendo un valor medio de puertos designados (que forman parte del árbol) de cuatro por conmutador podríamos alojar hasta 4096 conmutadores y casi 50.000 si llegamos a 6 puertos por conmutador.

Si fuera necesario, podría reconfigurarse el número de bits de cada nivel para permitir profundidades mayores, pero nuestros evaluaciones demuestran que no son necesarios en la mayoría de los casos. Incluso podríamos tener niveles de tamaño variable.

Aunque todas las direcciones HLMAC ocupan 6 octetos se consideran de tamaño variable ya que parte de la dirección (los últimos octetos) quedarán inutilizados (todo '0's) salvo en los nodos alojados en el último nivel. Por simplicidad, manejaremos direcciones 'recortadas', a las que se les han eliminado los últimos octetos (nulos) que no resultan relevantes tanto en las figuras como en las explicaciones posteriores.

Vamos a utilizar un esquema de notación basado en la separación de octetos (niveles) mediante puntos, de un modo similar al formato de direcciones IP por su familiaridad. Además, se referirán únicamente a los 46 hábiles una vez que los dos menos significativos (individual/grupo global/local) quedan fijados a '0' y '1' respectivamente.

Como ejemplo, en la tabla siguiente se representa la dirección HLMAC '1.2.3.4.0.0' ó '1.2.3.4' en forma simplificada.

Tabla 3. Ejemplo de dirección HLMAC: '1.2.3.4' (bits en forma canónica, *LSB* primero)

01100000	**01000000**	**11000000**	**00100000**	**00000000**	**00000000**
1	2	3	4	0	0

3.4.1.2 Asignación de la dirección HLMAC

Se propone utilizar la numeración de puertos de los conmutadores como elemento clave para la asignación de identidades HLMAC de manera que dados dos conmutadores A y B, conectados directamente, y pertenecientes a la misma rama del árbol de manera que B sucede a A en el mismo (está a un salto mas de distancia de la raíz), la dirección HLMAC de B se construye simplemente añadiendo el número del puerto del conmutador A que lo une a B a la propia dirección HLMAC de A, en realidad se trata de un proceso de delegación en el que A indica a B cual será su dirección jerárquica. De esta manera se elimina la necesidad de coordinar la numeración y, por tanto, el proceso es perfectamente distribuible.

La Ilustración 18 muestra un ejemplo del procedimiento de asignación de HLMAC a los nodos de una red. Por ejemplo, si tomamos el nodo D3, su dirección será '32.7.8' debido a que su predecesor en el árbol (D2) tiene la dirección '32.7' y se conecta a D3 a través de su puerto 8; a su vez D2 toma la dirección de su predecesor (D1), en este caso '32' y le añade el puerto que une D1 a D2 (el número 7).

Finalmente D1 toma su dirección directamente de la raíz a la que se conecta a través del puerto '32' del nodo raíz.

Con este esquema podemos manejar conmutadores con hasta 255 puertos, lo que es más que suficiente, prácticamente siempre, salvo que se trate de conmutadores finales (o frontera a los que se conecten equipos de usuario). En el primer nivel solo se admitirían 63 nodos, al tener reservados dos bits mas la propia dirección del nodo raíz.

Si llegado el caso, un conmutador adquiere una dirección de tamaño máximo (sexto nivel), el proceso no puede continuar aun en el caso de que el árbol sea más profundo. El propio conmutador detecta que está en el límite de aplicación del sistema de direccionamiento y pasaría a comportarse como frontera del dominio HURP, dando por finalizado el proceso de delegación de direcciones HLMAC. Aquellos otros conmutadores situados por debajo en el árbol tomarán los conmutadores frontera HURP como raíces de su propio árbol de expansión estándar, es decir, daría lugar a una isla Ethernet estándar colgando del nodo frontera del dominio HURP. La comunicación entre los dominios estándar y HURP requeriría en este caso el encapsulado de las tramas en la frontera. De hecho, un conmutador HURP, que no siendo raíz, no reciba por delegación su propia dirección HLMAC, se comportaría exactamente igual que si fuera un conmutador estándar que no reconoce el protocolo HURP.

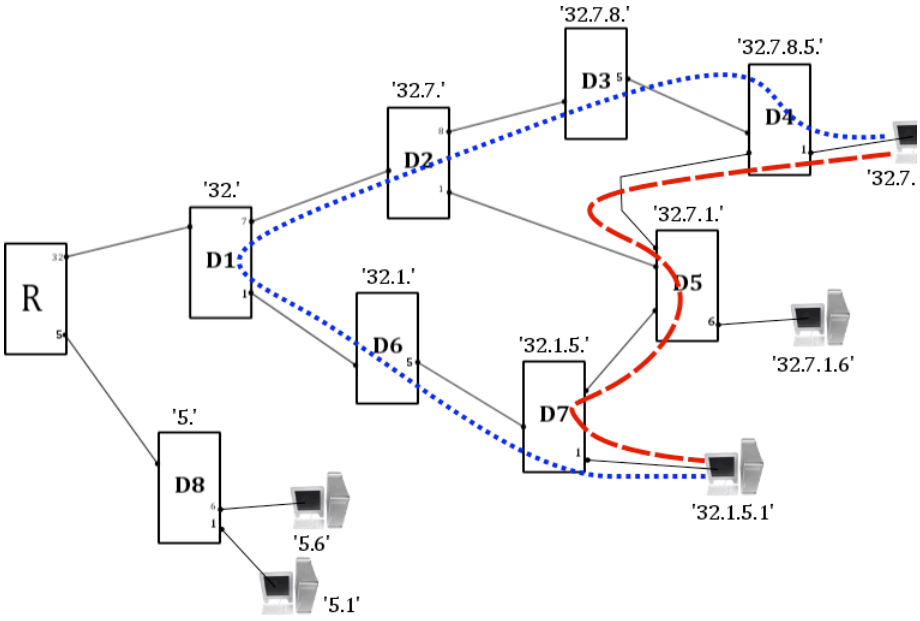


Ilustración 18. Ejemplo de asignación de direcciones HLMAC

3.4.2 El protocolo de árbol de expansión combinado CSTP (*Combined* STP)

Denominamos CSTP al conjunto formado por el protocolo estándar RSTP encargado de la construcción y del mantenimiento del árbol de expansión junto con las extensiones que permiten aprovechar el diálogo entre nodos para la construcción del árbol RSTP con la asignación mediante delegación del direccionamiento jerárquico HLMAC.

Las tramas de control RSTP (BPDU) se extienden con un nuevo campo (6 octetos) que permite alojar la dirección HLMAC delegada. De esta manera, durante el proceso de construcción del árbol un nodo recibirá varias HLMAC delegadas de sus vecinos y validará la que corresponda al puerto que RSTP determine como puerto raíz. Este proceso se repite en el tiempo (basado en un temporizador

hello_time) de manera que se detectan los posibles cambios topológicos y se reconfiguran tanto el árbol como las HLMAC cuando se necesite. Un nodo considera que su HLMAC es válida (se puede usar para el reenvío) cuando el protocolo original RSTP establece su puerto raíz y el correspondiente puerto designado en su predecesor en el árbol, entonces, puede delegar su propia HLMAC a sus hijos (sucesores).

En definitiva, se trata de un proceso en cascada desde el nodo raíz a partir del cual se van definiendo los correspondientes pares de puerto designado (en el padre) y puerto raíz (en el hijo) que extiende paulatinamente el árbol y el direccionamiento jerárquico en paralelo hasta los nodos más alejados.

Al igual que con la migración de STP a RSTP es necesario definir un nuevo identificador de versión de protocolo de árbol de manera que cuando se reciban las BPDU, los nodos sepan claramente cuál es el comportamiento que deben ejecutar eligiendo para cada par de puertos la combinación de mayor precedencia posible (si ambos hablan HURP primero, luego si ambos hablan RSTP y, en última instancia, se usaría STP, la norma básica).

3.4.3 Encaminamiento y reenvío en HURP

HURP es un protocolo de encaminamiento y reenvío, basado en vectores de distancia, con prohibición de giros para el control de bucles y direcciones jerárquicas.

Los nodos HURP intercambian sus tablas de vectores de distancia utilizando una topología activa construida en base al encaminamiento *Up/Down* con prohibición de giros mejorada con el uso de direcciones jerárquicas. Se permite el uso de enlaces transversales (fuera del árbol) en tanto en cuanto no violen las reglas que prohíben determinados giros y gracias a la información extra aportada por la jerarquía del direccionamiento, se pueden incluso permitir parte de esos giros prohibidos para el reenvío de las tramas destinadas a la rama en que desemboca el giro. Una vez que una trama alcanza la rama destino, no es necesario encaminar, se puede reenviar directamente simplemente descodificando su HLMAC, ni siquiera es necesario el aprendizaje.

3.4.3.1 Plano de control

Los mensajes de control que intercambian los nodos HURP son similares a los de un protocolo de vectores de distancia como RIP (*Routing Internet Protocol*), pero con intervalos de tiempo menores (por debajo de un segundo) con el objetivo de mejorar los tiempos de reconfiguración en caso de fallos o cambios topológicos. La métrica puede ser tan simple como el número de saltos o una métrica similar a la usada por la norma 802.1D, con costes inversamente proporcionales al régimen binario de los enlaces.

Cada nodo HURP selecciona rutas que atraviesen el árbol cruzando de unas ramas a otras siempre y cuando mejoren el coste frente al reenvío directo vía árbol y no incurran en un giro prohibido. Los nodos construyen sus propias tablas a partir de la información que reciben de sus vecinos aplicando el algoritmo de Bellman-Ford. Sin embargo, a la hora de reenviar sus tablas a sus vecinos deben filtrar aquellas rutas que puedan resultar en un giro prohibido (el identificador del nodo seleccionado y el del vecino son ambos mayores que el propio) de manera que no todos los vecinos reciben la misma información. También se filtran las rutas aprendidas del propio vecino (clásico corte por horizonte dividido).

Se utiliza un protocolo tipo RIP no solo por su simplicidad sino porque al combinarlo con la prohibición de giros la topología activa resulta libre de bucles y, por tanto, está libre del problema de

la cuenta a infinito que lastra este tipo de protocolos. A cambio, para acelerar la convergencia, paso a paso, el proceso de intercambio de información debe acelerarse.

Al combinar este protocolo con la posibilidad de habilitar los giros prohibidos que terminan en la rama destino de la trama se consigue una mejora en el rendimiento sobre el protocolo clásico *Up/Down* muy importante.

Si se produce un fallo en un enlace o en un nodo de la red, el árbol de expansión puede verse afectado y necesitar un proceso de reconfiguración (dramático si es el propio raíz el afectado). La reconfiguración en HURP descansa sigue las mismas reglas que en RSTP, de hecho, mientras RSTP no reconstruya y active de nuevo el árbol no dispondremos de direcciones jerárquicas útiles. Mientras que RSTP descarta las direcciones MAC que tenga aprendidas en los nodos, HURP debe invalidar las direcciones HLMAC y purgar todas las rutas afectadas. El reenvío basado en HLMAC se detiene inmediatamente en los puertos afectados hasta que disponen de una nueva HLMAC válida. Mientras dura este proceso solo se pueden reenviar tramas siguiendo las reglas básicas de RSTP (vía árbol).

La dependencia del direccionamiento HLMAC de RSTP puede parecer un factor de riesgo importante, sin embargo, solo en el caso de un fallo en la raíz del árbol el proceso de reconfiguración es realmente costoso. La mayoría de los fallos no tienen porqué afectar a ninguna dirección, o solo a un pequeño subconjunto (la sub-rama del árbol que deba reconstruirse), además, RSTP ya incluye mecanismos de reconfiguración rápida que minimizan la posible indisponibilidad temporal de direcciones HLMAC.

3.4.3.2 Plano de usuario

El reenvío de tramas en HURP dispone de dos modos de operación:

- Mediante un algoritmo de camino más corto del tipo vector de distancias aplicado sobre toda la topología (con excepción de los giros prohibidos). Por tanto, utiliza todos los enlaces de la red aunque no todas las combinaciones sean posibles como en un *shortest path* clásico. Se trata del modo predeterminado o modo por defecto.

- Sin tablas de encaminamiento, simplemente reenviando por el árbol (en sentido ascendente hasta alcanzar un nodo común entre los nodos origen y destino, y luego descendente por la rama que aloja al destino). En el segundo caso, el camino puede incluir el ascenso por el árbol hasta el nodo raíz si se da el caso de que los nodos origen y destino se encuentren en ramas disjuntas del árbol.

A continuación se describe el proceso correspondiente al algoritmo de reenvío que ejecutarán los conmutadores HURP para el reenvío de tramas en el modo predeterminado. El algoritmo opera de la siguiente manera:

Si las direcciones HLMAC del destino y del propio están incluidas una en la otra indistintamente (se encuentran en la misma rama del árbol):

- Si la dirección HLMAC del propio conmutador está incluida (es un prefijo de ella) en la dirección HLMAC del destino de la trama la trama es simplemente reenviada por el puerto designado correspondiente (hacia abajo por la rama del árbol que corresponda).

- Si la dirección HLMAC del destino está incluida en la del propio conmutador (es un prefijo de ella), la trama es reenviada por el puerto raíz (hacia arriba en el árbol).

En ninguno de los casos es necesario que haya aprendizaje como en una conmutador transparente, una simple operación lógica de mapeo de las HLMAC permite tomar la decisión.

Si la dirección destino HLMAC se encuentra en otra rama del árbol, se consulta la tabla de rutas para determinar el siguiente salto a seguir, sea éste a través de un enlace del árbol o a través de un enlace entre ramas (*cross-link*). Las tablas de rutas pueden referenciar caminos individualizados o agregar destinos en una sola entrada. Por ejemplo, para el envío de una trama desde el host '32.7.8.5.1' al host '32.1.5.1', que no comparten rama, si el nodo D4 conoce la ruta hasta D7 vía D5 (línea roja punteada) reenviará la trama hacia D5, de ahí a D7, que la entregará al destino directamente, en total, 4 saltos de origen a destino.

Si no se conoce otra alternativa mejor la trama es reenviada por el puerto raíz (hacia arriba en el árbol). En el mismo caso del párrafo anterior, si los nodos D4 primero y D2 más tarde no conocen una ruta hacia D7 la trama será reenviada por el árbol hasta D1, que comparte rama con el destino, desde ahí descenderá por la rama vía D6 y D7 hasta el destino (línea azul discontinua), en total, 7 saltos.

3.4.4 Compatibilidad con conmutadores estándar y autoconfiguración

La interoperabilidad entre los conmutadores HURP y los conmutadores estándar 802.1D se consigue mediante la autoconfiguración de un núcleo de conmutadores HURP conectados directamente, al que se unen el resto de conmutadores estándar (o conmutadores HURP aislados) mediante la formación de árboles independientes enraizados en alguno de los conmutadores HURP frontera. Las tramas que atraviesan la frontera entre los dominios HURP y un árbol estándar deben ser des/encapsuladas con las correspondientes direcciones HLMAC de los nodos frontera HURP.

El proceso de autoconfiguración del núcleo HURP se basa en el protocolo de árbol combinado (CSTP) descrito previamente y recogido en la Ilustración 19 (que muestra la topología original con indicación del tipo de cada uno de los nodos y, la Ilustración 20, con la topología resultante con el núcleo HURP generado y los distintos árboles RSTP (clásicos) que derivan de él para construir una topología activa única y conectada. Los nodos HURP frontera se auto-configuran como nodos de máxima prioridad (utilizando los bits de prioridad de su dirección MAC) y se convierten de esta manera en nodos raíz de los árboles estándar. También se destacan algunos nodos HURP que, por encontrarse aislados (solo tienen enlace directo con nodos estándar), 'rebajan' su comportamiento al de un nodo estándar más.

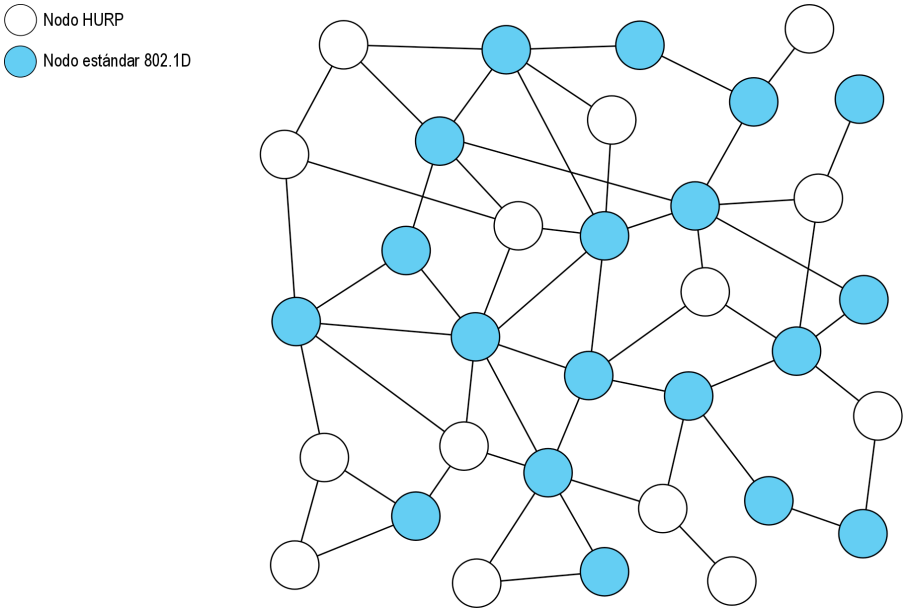


Ilustración 19. Autoconfiguración de HURP (topología arbitraria original)

Dentro del núcleo HURP la única restricción al encaminamiento la imponen los giros prohibidos al aplicar la reglas HURP (*up/down* mejoradas) mientras que en los árboles estándar se aplica la prohibición clásica de enlaces determinada por R/STP.

Para el reenvío de una trama entre dos nodos estándar que se encuentren en árboles disjuntos se utiliza el núcleo HURP a modo de red de interconexión. La trama original se transmite desde el nodo origen hasta la raíz de su árbol (un nodo frontera HURP) donde es encapsulada con la dirección HLMAC del nodo HURP frontera al que debe dirigirse, éste, a su vez, procede a desencapsular la trama y entregarla en su árbol estándar para que siga la rama que corresponda hasta el destino.

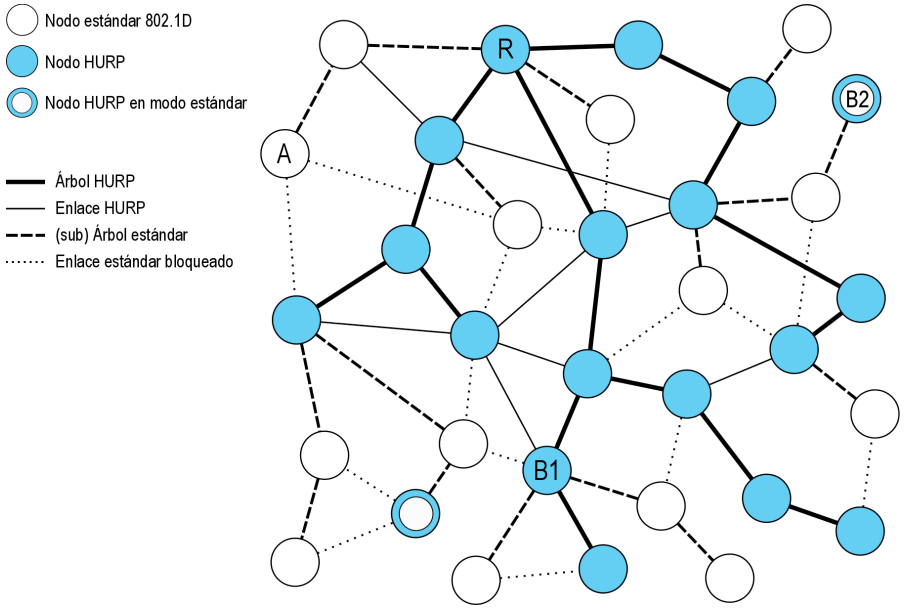


Ilustración 20. Autoconfiguración de HURP (núcleo HURP y subárboles estándar)

El proceso se ilustra en la Ilustración 21, donde los nodos que conforman el núcleo HURP se representan en azul y se indica mediante el identificador R qué nodo HURP actúa de raíz así como en trazo grueso los enlaces del árbol HURP. En esta red se pretende transmitir una trama Ethernet entre los sistemas finales G1 y G2 (por tanto G1 y G2 utilizan direcciones globales). G1 está conectado directamente a B1 mientras que G2 está conectado a un nodo estándar que depende del nodo HURP frontera B2 (B1 y B2 tienen tanto una dirección global como una dirección local HLMAC).

Cuando el equipo G1 envía una trama *unicast* destinada a la dirección global G2, está es entregada al nodo B1 que la encapsula para darle entrada al dominio HURP. Dado que, en principio B1 desconoce la situación de G2, utiliza como nueva dirección destino la dirección de difusión y la trama será difundida por el árbol HURP. Cuando una de las copias llegue al nodo frontera B2, éste desencapsula la trama original, aprende la relación G1-B1 para usos futuros y reenvía la trama original hacia G2 (los nodos estándar en el camino aprenderán la localización de G1 en este proceso).

Si ahora el equipo G2 responde con una nueva trama *unicast* dirigida de vuelta a G1, la trama será reenvidada hasta B2 donde debe ser encapsulada para atravesar el dominio HURP. En este caso, la dirección destino será la del nodo B1 (puesto que B2 ya conoce la asociación B1-G1) y la trama puede ser encaminada utilizando el mejor camino posible entre B2 y B1. De nuevo en B1, se procede a desencapsular y entregar la trama original a G1. Además, el nodo B1 aprende la asociación B2-G2 y, en el futuro, no necesitará recurrir a la difusión de nuevo para enviar tráfico de G1 a G2.

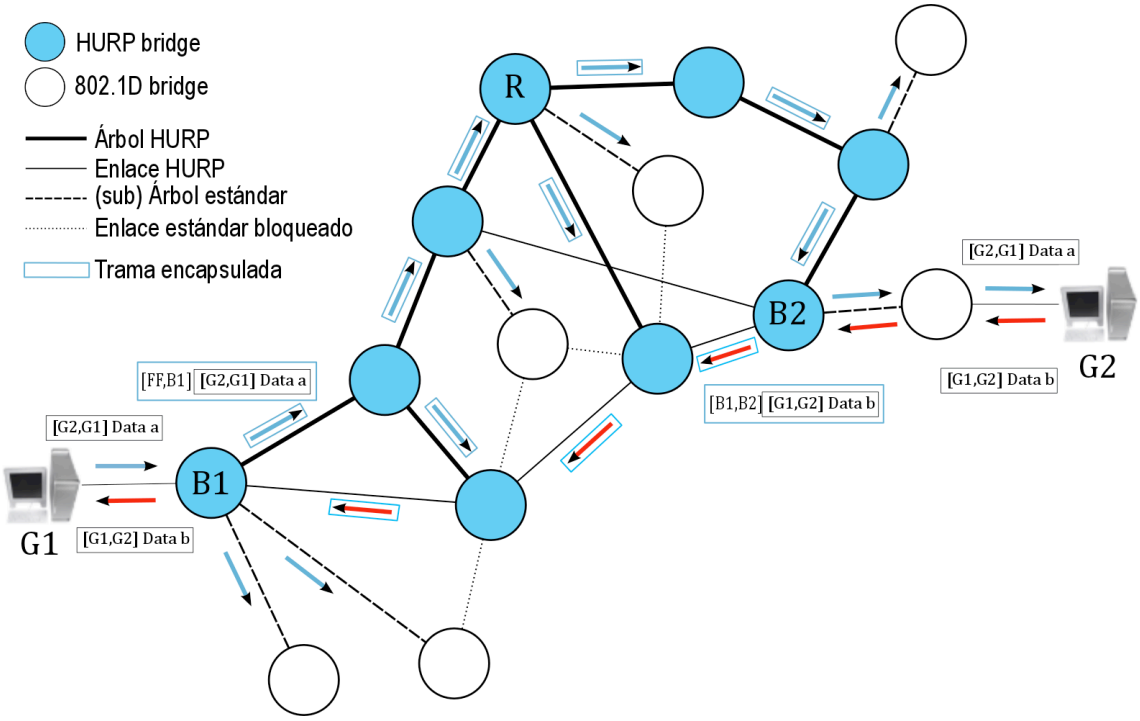


Ilustración 21. Transmisión entre nodos estándar vía el núcleo HURP

3.4.5 Análisis de complejidad

La complejidad computacional de HURP en el caso peor es polinomial con N, $O(N.d_{max}^2)$, dónde N es el número de nodos de la red y d_{max} el grado máximo de cualquiera de los nodos de la misma.

El protocolo HURP combina la operación de un algoritmo de vectores de distancia y RSTP y necesita realizar los siguientes cuatro procesos:

- Construcción del árbol RSTP: STP es básicamente un vector de distancias con un único nodo como objetivo (la raíz del árbol). Los vectores de distancia al raíz se intercambian entre nodos vecinos y, por tanto, su complejidad es $O(N.d_{max})$

- Asignación de la dirección HLMAC: las direcciones locales se asignan en función del estado de los puertos de cada nodo. En el peor caso, un nodo asignará d_{max} direcciones (una por vecino), por lo que de nuevo su complejidad es $O(N.d_{max})$

- Cálculo de giros prohibidos: los giros prohibidos se calculan comparando el valor de las direcciones locales (jerárquicas) de los nodos. En un nodo con d vecinos, hay $d(d-1)/2$ giros posibles que evaluar, por lo que la complejidad será $O(d^2)$ para un nodo y $O(Nd_{max}^2)$ para la red en su conjunto.

- Cálculo de los caminos mínimos: cada nodo calcula su tabla de vectores y la envía a sus vecinos filtrando para cada vecino las rutas que incurran en giros prohibidos, cada nodo recibirá un máximo de d tablas (una por vecino) y su complejidad será equivalente a la del algoritmo de Bellman-Ford, esto es, $O(N.d_{max})$

3.4.6 Evaluación

En esta sección se presenta una comparativa con los resultados de rendimiento obtenidos mediante simulación de HURP, *Up/Down* (U/D), STP y SP (*Shortest Path*). Se evalúa el rendimiento global (*throughput*) y el tamaño medio de los caminos resultantes. Asimismo, se presentan resultados sobre el número total de giros prohibidos con el doble objetivo de evaluar la ventaja cuantitativa que proporciona el direccionamiento local jerárquico aplicado a la prohibición de giros y justificar la mejora de rendimiento global obtenida.

3.4.6.1 Metodología de evaluación

Se ha optado por realizar simulaciones basadas en modelos de flujos desarrollado en Python [PYTHON] en lugar de trabajar al nivel de paquete, con ello se pretende evaluar redes de gran tamaño durante tiempos suficientemente grandes que resultan inabordables con un simulador clásico de paquetes.

Se evalúan distintos tipos de topologías de red, aleatorias y regulares, con distintos tamaños y parámetros de relación como el grado medio de los nodos. En el caso de las topologías aleatorias se utilizan los modelos de generación de redes aleatorio puro (Waxman) y libre de escala (Barabasi-Albert).

Todos los enlaces evaluados tienen la misma capacidad y se usa el número de saltos como métrica de costes (todos los enlaces tienen el mismo coste, uno).

Para poder comparar específicamente nuestros resultados con los de otros protocolos como TP y TBTP se han incluido también en el análisis ciertas topologías y condiciones concretas mencionadas en las respectivas referencias.

Dado que el protocolo U/D presenta un comportamiento bastante dependiente del nodo raíz elegido en el proceso de construcción del árbol para posteriormente asignar los identificadores correctamente, en cada topología simulada se repiten las ejecuciones variando el nodo elegido como raíz y se presentan los valores medios resultantes.

Para cada topología primero se calcula el árbol de expansión y se asignan los identificadores locales en consecuencia; luego se calculan los giros prohibidos que resultan de aplicar las reglas correspondientes a STP, U/D y HURP y se calculan las rutas (mejores caminos) resultantes sobre la topología activa que haya resultada en cada caso. Finalmente, se presentan los resultados obtenidos en términos de porcentaje de giros prohibidos, tamaño del camino medio y rendimiento máximo.

3.4.6.2 Porcentaje de giros prohibidos

El porcentaje de giros prohibidos se evalúa tanto en redes regulares como aleatorias, y en ciertas topologías concretas que se utilizan en la práctica en escenarios específicos. No se incluyen en la comparativa los resultados de STP debido a que el número de giros prohibidos es muy alto (entre el 60 y el 95% según el grado y tipo de topología) y distorsiona la ventaja relativa entre HURP y U/D.

3.4.6.2.1 Topología regulares: mallas tridimensionales

Se han evaluado mediante simulación mallas cúbicas (tridimensionales) con 3, 4, 5 y 6 nodos por lado (por tanto, mallas de N=3*3*3=27, N=4*4*4=64, N=5*5*5=125 y N=6*6*6=216 nodos), realizando tantas ejecuciones para cada topología como nodos, en las que se elegía un nodo diferente como raíz cada vez.

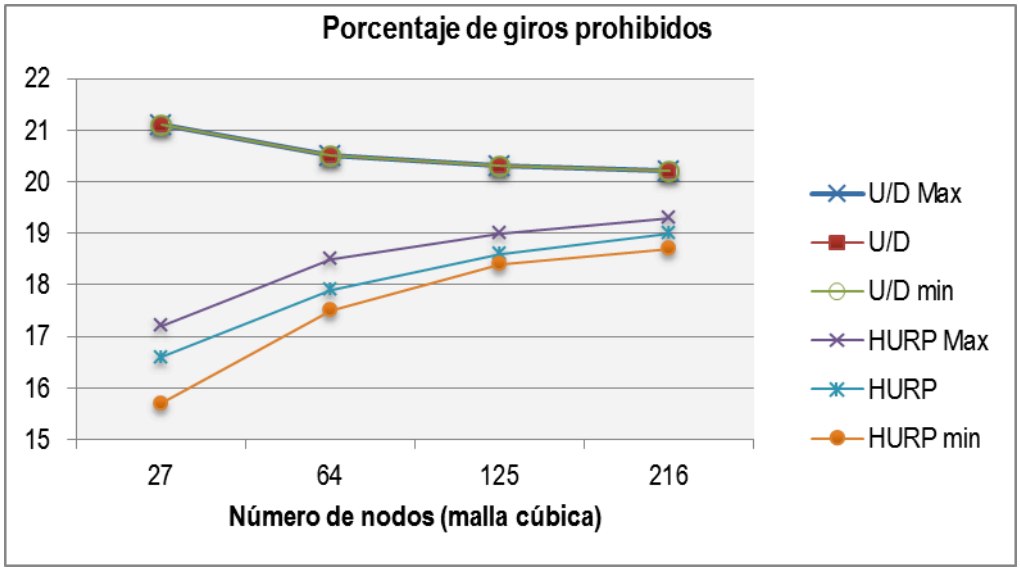


Ilustración 22. Porcentaje de giros prohibidos en mallas cúbicas regulares

HURP mejora la fracción de giros prohibidos (0,16-0,19 en media) de U/D que, ya de por sí, ofrecía porcentajes pequeños (entre 0,2 y 0,21) gracias al mecanismo añadido que utiliza la dirección jerárquica para permitir nuevos giros prohibidos en el protocolo original. Como era de esperar, la

ganancia se reduce conforme aumenta el tamaño de la topología ya que proporcionalmente aumenta más el número de giros totales.

3.4.6.2.2 Topologías aleatorias

Para evaluar el efecto que sobre el porcentaje de giros prohibidos tienen parámetros como el tamaño de la red, el grado medio o la distribución estadística del grado de los nodos se han simulado distintas topologías aleatorias de 120 nodos de grado fijo y de grado aleatorio (modelo de Waxman). Además, se han simulado varios grupos de topologías aleatorias puras (Waxman), sin escala (Barabasi-Albert) y de grado fijo, con tamaños desde 16 hasta 128 nodos, generadas mediante la herramienta BRITE [BRITE].

En el caso de las topologías Waxman de 120 nodos y distinto valor de grado medio, se han evaluado 40 topologías diferentes. Los resultados obtenidos pueden verse en la Ilustración 23. HURP mejora los resultados de *Up/Down* entre un 5-10% en todos los casos. Además, es de destacar que los valores máximo y mínimo en HURP se encuentran bastante cercanos a la media por lo que la dependencia con respecto al nodo raíz elegido es baja.

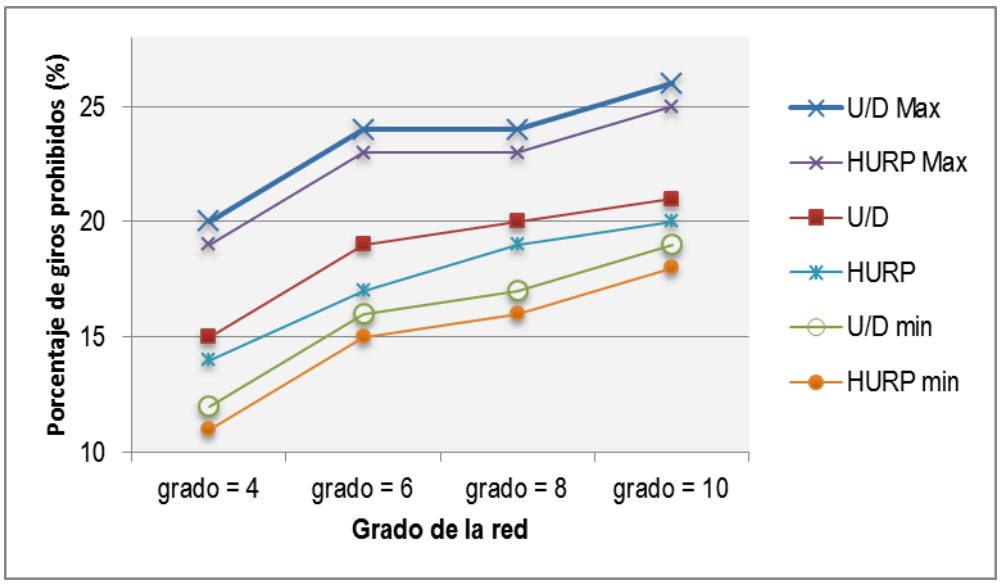


Ilustración 23. Porcentaje de giros prohibidos en HURP, Up/Down y STP (solo tabla)

En el caso de las topologías de grado fijo, la Ilustración 24 muestra los resultados obtenidos evaluando 40 topologías diferentes y los mismos grados que en el caso anterior. Como puede verse los resultados son consistentemente peores para todos los protocolos, incluso peores que en redes regulares. En cualquier caso, los valores máximos obtenidos para HURP se encuentran siempre por debajo del valor medio en Up/Down y, al igual que ocurría con las topologías de grado variable, se observa también el aumento claro del porcentaje de giros prohibidos con el grado medio de la topología. En realidad, se trata del grupo de topologías menos representativo de todas las estudiadas ya que son difíciles de encontrar en la práctica.

Finalmente, en el caso de las topologías aleatorias generadas según el modelo sin escala (Barabasi-Albert) se han simulado topologías con tamaños crecientes desde 32 hasta 256 nodos (30 en cada caso) con valores de grado de 4, 6 y 8 para cada caso. Los resultados obtenidos se muestran en la Ilustración 25.

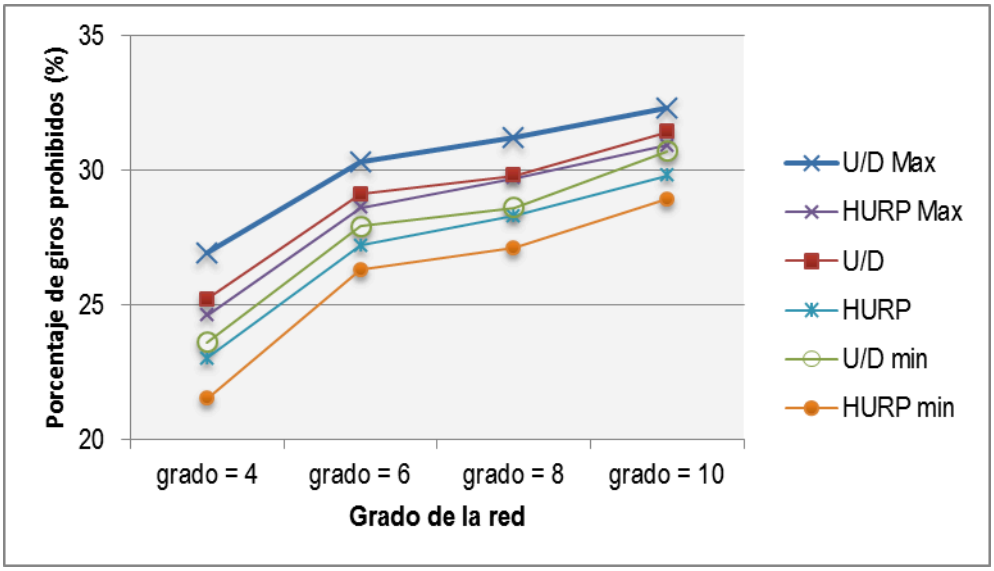


Ilustración 24. Porcentaje de giros prohibidos en topologías Waxman de grado fijo

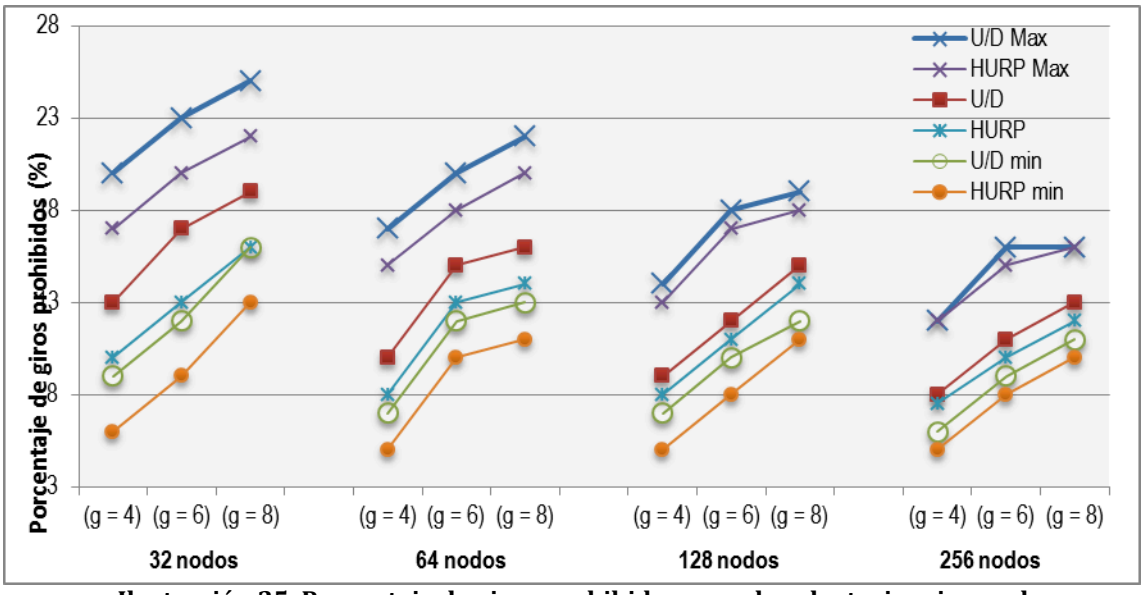


Ilustración 25. Porcentaje de giros prohibidos en redes aleatorias sin escala

Puede verse claramente como los valores máximos, medio y mínimo, son siempre mejores para HURP, crecen con el grado de la topología y, sin embargo, decrecen con el tamaño de la red. Esto es debido a que los nodos de grado alto terminan situándose cercanos a la raíz y, por tanto, producen menos giros prohibidos que en el caso de grado aleatorio. En todo caso, el valor máximo para ambos protocolos se encuentra claramente por debajo del 30% de giros prohibidos.

La Tabla 4 y la Tabla 5, presentan los resultados comparados incluyendo otros protocolos de giros prohibidos como TP y TBTP. La Tabla 4 con topologías de 120 nodos y nodos de grado fijo con valores desde cuatro hasta 10. Por su parte, la Tabla 5, con topologías de tamaño variable (desde 16 hasta 128 nodos) y grafo fijo de valor cuatro en todos los casos. Los valores obtenidos para U/D y STP nos sirven para validar los resultados mostrados al comienzo de la sección. De la tabla podemos concluir que HURP proporciona valores de giros prohibidos similares a TBTP, pero se trata de un protocolo distribuido y mucho más simple que éste.

Tabla 4. Fracción de giros prohibidos en topologías de 120 nodos y distinto grado fijo

Grado (d)	Up/Down	HURP	TBTP	TP	SpanningTree
4	0,25	0,23	0,23	0,21	0,73
6	0,29	0,27	0,27	0,23	0,86
8	0,30	0,28	0,28	0,25	0,91
10	0,31	0,30	0,29	0,26	0,93

Tabla 5. Fracción de giros prohibidos en redes de tamaño variable y grado fijo (valor 4)

Nº nodos	Up/Down	HURP	TP	SpanningTree
16	0,27	0,20	0,23	0,72
32	0,26	0,21	0,23	0,73
64	0,25	0,22	0,22	0,73
128	0,25	0,23	0,21	0,73

En el caso de la Tabla 5 no aparecen los resultados de TBTP ya que la referencia en cuestión solo proporciona valores para redes de grado 8 (muy poco realistas). Puede verse como HURP mejora los resultados de TP en redes pequeñas pero empeora según aumenta el tamaño aunque siempre mejorando los valores de U/D.

Como resumen del estudio del porcentaje de giros prohibidos podemos concluir que:

- El tipo de topología tiene una gran influencia en el valor nominal de todos los protocolos pero no produce un cambio relativo en el rendimiento comparado entre ellos.
- Tanto U/D como HURP presentan una dependencia con la raíz elegida limitada (valores máximo y mínimo) y siempre dentro de límites razonables. U/D plantea "preseleccionar" candidatos adecuados a ser raíz para limitar este efecto. En HURP no parece necesario ya que incluso en redes sin escala (con gran variabilidad de grado) los nodos de grado alto se colocan naturalmente en posiciones cercanas a la raíz independientemente de cuál sea la raíz elegida.
- HURP proporciona resultados consistentemente mejores que U/D para redes de grado medio entre 4 y 8 y tiende a los mismos valores para grados mayores (de nuevo, redes de grado medio 8 o más resultan poco prácticas y difíciles de encontrar en la realidad).

3.4.6.3 Rendimiento (*throughput*)

Utilizando las mismas topologías mencionadas en la sección anterior se han realizado medidas de rendimiento de la red. En la evaluación se asume que cada nodo de la red tiene un cliente conectado que establece una sesión (flujo unitario) con otro cliente conectado a cada uno de los demás nodos de la red, en total (*N-1*) flujos por cliente (nodo). Las simulaciones se realizan tomando cada nodo de la red como raíz, por tanto, *N* iteraciones, para evitar el posible efecto de la elección en el resultado final.

Definimos el rendimiento de la red, basándonos en el modelo de flujos propuesto, como el de aquel enlace que se constituye en el cuello de botella de la misma (soporta más flujos unitarios) una vez que se han calculado las rutas para cada posible algoritmo y par de nodos origen/destino. Tomamos como valor objetivo el obtenido para un protocolo SP (mejor caso posible) y referenciamos

cada protocolo frente a ese valor objetivo, por tanto, se trata de una medida de rendimiento relativo. Un protocolo SP puede no resultar óptimo en función de la distribución del tráfico en la red, en nuestras simulaciones se producen rendimientos relativos superiores al 100% debido al sistema de desempate que utiliza el algoritmo de Dijkstra cuando existen múltiples caminos de igual coste, es decir, no realiza balanceo de carga.

3.4.6.3.1 Topologías regulares

Se han evaluado mediante simulación mallas cúbicas (tridimensionales) con 3, 4, 5 y 6 nodos por lado (por tanto, mallas de N=3*3*3=27, N=4*4*4=64, N=5*5*5=125 y N=6*6*6=216 nodos), realizando tantas ejecuciones para cada topología como nodos, en las que se elegía un nodo diferente como raíz cada vez.

U/D y SP producen resultados similares ya que al existir una gran variedad de caminos alternativos de similar coste U/D siempre encuentra un camino alternativo de igual coste. HURP mejora los resultados de U/D entre un 20% y un 5%, menor cuanto más grande es la red. Esta mejora por encima del potencial objetivo tiene que ver de nuevo con un efecto de balanceo de carga que no se tiene en cuenta en SP.

Tabla 6. Porcentaje de *Rendimiento Relativo* frente a SP

Dimensión	Nº nodos	Grado (d)	ShortestPath	HURP	Up/Down	SpanningTree
3x3x3	27	4	100	104,31	100	36,58
4x4x4	64	4,5		110,8	100	28,81
5x5x5	125	4,8		116,95	100	24,17
6x6x6	216	5		121,84	100	21,57

3.4.6.3.2 Topologías aleatorias

Se han simulado topologías aleatorias de tipo Waxman con grado variable y 120 nodos, topologías de grado fijo y topologías sin escala (Barabasi) de tamaño y grado medio variables.

La Tabla 7 muestra los valores obtenidos para topologías Waxman de 120 nodos con grado medio variando desde 4 hasta 10.

Tabla 7. Rendimiento relativo a SP (%) en redes Waxman de 120 nodos y grado variable

Nº nodos	Grado (d)	ShortesPath	HURP	Up/Down	SpanningTree
120	4	100	81	57	12,9
120	6	100	86	65	7,4
120	8	100	88	69	5,6
120	10	100	96	73	4,9

HURP mejora claramente los resultados de U/D pasando del orden de un 60% de SP al 80%. La mejora es aproximadamente constante con el grado medio de la red pero el rendimiento relativo crece de manera monótona para ambos protocolos y se acerca al 100% de SP en el caso de HURP y grado 10 (nuevamente hay que incidir que topologías de grado medio muy alto son poco significativas y difíciles de encontrar en la realidad). Esta mejora se debe a que al aumentar el grado aumenta también el número de caminos óptimos y, por tanto, el efecto de prohibir algunos de ellos es cada vez menos apreciable.

En el caso de las topologías de grado fijo (de valor 4), los resultados obtenidos siguen un comportamiento similar al comentado para el caso del porcentaje de giros prohibidos: HURP produce rendimientos relativos del orden de un 20% nominal mejor que U/D independientemente del tamaño de la topología, sin embargo, ambos protocolos degradan claramente su rendimiento frente a SP según crece el tamaño de la red. Al tener el mismo grado todos los nodos, el número de giros prohibidos de la topología crece más rápidamente que el número de giros totales y los caminos resultantes no son óptimos. La Tabla 8 muestra los valores obtenidos.

Tabla 8. Rendimiento relativo a SP en topologías de grado fijo (valor4) y tamaño variable

Nº nodos	ShortesPath	HURP	Up/Down	SpanningTree
16	100	88	66	42
32	100	79	55	36
64	100	65	41	26
128	100	47	25	15

Por último, la Tabla 9 muestra los valores obtenidos con topologías aleatorias sin escala (Barabasi-Albert) tanto para tamaños como para grados medios variables, entre 32 y 256 nodos y grados medios de 4, 6 y 8 aproximadamente. Puede observarse como HURP proporciona rendimientos cercanos a SP en todos los casos, además, mejora los resultados de U/D desde un 10% nominal para redes pequeñas hasta casi un 20% en las de mayor tamaño. Es destacable que no hay variaciones apreciables con el grado medio de la red. Nuevamente, la explicación tiene que ver con el hecho de que los nodos de mayor grado son "empujados" hacia arriba en el árbol por estar muy conectados, independientemente de cual sea la raíz elegida. Los valores superiores al 100% se explican por efectos de balanceo de carga colaterales a la prohibición de giros y que no se contemplan en SP.

Tabla 9. Rendimiento relativo a SP para topologías aleatorias sin escala de tamaño y grado variables

Nº nodos	Grado (d)	ShortesPath	HURP	Up/Down	SpanningTree
32	3,81	100	97	89	29
	5,63		95	91	17
	7,38		94	91	12
64	4		99	88	26
	5,81		97	86	14
	7,69		101	90	10
128	3,95		102	81	21
	5,90		101	83	11
	7,84		109	93	9
256	3,98		103	80	19
	5,95		104	82	10
	7,92		97	81	6

3.4.6.4 Camino medio

El tercer parámetro a considerar en la comparativa es el tamaño medio de los caminos seleccionados por cada protocolo para los mismos casos de estudio de los dos parámetros anteriores. Se asume el mismo modelo de distribución de tráfico uniforme dentro de la red, con un flujo por nodo, con destino a cada uno de los demás nodos de la red.

3.4.6.4.1 Topologías regulares: mallas tridimensionales

Los resultados obtenidos para las mallas cúbicas regulares (tridimensionales) coinciden para HURP, U/D y SP. Esto se debe a que al tratarse de topologías regulares siempre existen múltiples alternativas de igual coste (igual tamaño) y prohibir alguna de ellas no reduce la capacidad de alcanzar el destino por otra ruta del mismo tamaño.

Tabla 10. Camino medio en mallas tridimensionales

Dimensión	Nº nodos	Grado (d)	ShortestPath	HURP	Up/Down	SpanningTree
3x3x3	27	4	2,77	2,77	2,77	4,27
4x4x4	64	4,5	3,81	3,81	3,81	6,1
5x5x5	125	4,8	4,84	4,84	4,84	7,9
6x6x6	216	5	5,86	5,86	5,86	9,69

3.4.6.4.2 Topologías aleatorias

La Tabla 11, Tabla 12 y Tabla 13 recogen los valores de camino medio obtenidos para topologías Waxman de tamaño fijo (120 nodos) y distintos grado medio, para topologías de grado fijo (valor 4) y topologías aleatorias sin escala (Barabasi-Albert) de tamaño y grado variables.

Tabla 11. Camino medio en topologías Waxman de 120 nodos y distintos grados medios

Nº nodos	Grado (d)	ShortesPath	HURP	Up/Down	SpanningTree
120	4	3,47	3,56	3,76	5,65
120	6	2,84	2,87	2,98	4,84
120	8	2,53	2,54	2,62	4,43
120	10	2,34	2,35	2,41	4,19

En todos los casos, HURP muestra valores de camino medio consistentemente mejores que U/D y muy cercanos a SP, prácticamente idénticos en el caso de las topologías sin escala. En cualquier caso, las variaciones entre HURP, U/D y SP con respecto a este parámetro no son grandes en ninguna de las topologías estudiadas. Quizás lo más significativo es que, de nuevo, las topologías de grado fijo obtienen los peores resultados y con mayores diferencias relativas.

Tabla 12. Camino medio en topologías de grado fijo (valor 4) y tamaño variable

Nº nodos	ShortesPath	HURP	Up/Down	SpanningTree
16	2,16	2,19	2,41	3,43
32	3,2	3,34	3,82	5,15
64	4,3	4,71	5,52	7,21
128	5,38	6,2	7,35	9,38

Tabla 13. Camino medio en topologías aleatorias sin escala de tamaño y grado variables

Nº nodos	Grado (d)	ShortesPath	HURP	Up/Down	SpanningTree
32	3,81	2,43	2,44	2,48	3,51
	5,63	2,07	2,07	2,09	3,20
	7,38	1,88	1,88	1,90	3,05
64	4	2,78	2,79	2,85	3,93
	5,81	2,40	2,40	2,44	3,78
	7,69	2,17	2,17	2,19	3,50
128	3,95	3,16	3,18	3,27	4,63
	5,90	2,7	2,71	2,76	4,30
	7,84	2,43	2,43	2,45	3,94

3.4.6.4.3 Topologías específicas

Para completar el estudio comparativo de rendimiento de HURP se plantea en este apartado el uso de determinadas topologías específicas que representan escenarios de uso muy concretos como son las denominadas *Floor* y Metro. La figuras siguientes muestran las topologías Floor, Floor+ utilizadas en el cableado de grandes edificios y el núcleo de una topología de red Ethernet metropolitana.

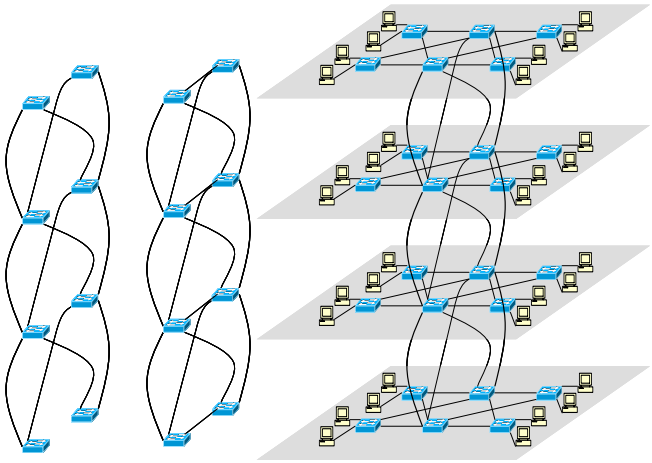


Ilustración 26. Topologías Floor y Floor+

Los resultados obtenidos en cuanto al porcentaje de giros prohibidos en las topologías Floor y Floor+:

Tabla 14. Porcentaje de giros prohibidos en topologías tipo floor*

Topología	SpanningTree	Up/Down	HURP
Floor	84	37	24
Floor+	75	31	21

Ni el rendimiento ni el camino medio muestra variaciones significativas entre U/D y HURP debido a que siempre hay un camino legal alternativo a los posibles caminos prohibidos y de igual coste.

En el caso de la topología metro, los resultados de la Tabla 15 se han obtenido asumiendo que hay cuatro clientes conectados a cada conmutador de acceso (plano inferior de la figura) y que establecen sesiones cuya distribución de destinos está aleatoriamente distribuida entre el resto de clientes.

Tabla 15. Resultados para la topología Metro-Ethernet

Parámetro	ShortesPath	HURP	Up/Down	SpanningTree
Giros prohibidos (fracción)	0	0,09	0,15	0,64
Rendimiento (%)	100	93	83	54
Camino medio	2,29	2,29	2,37	2,92

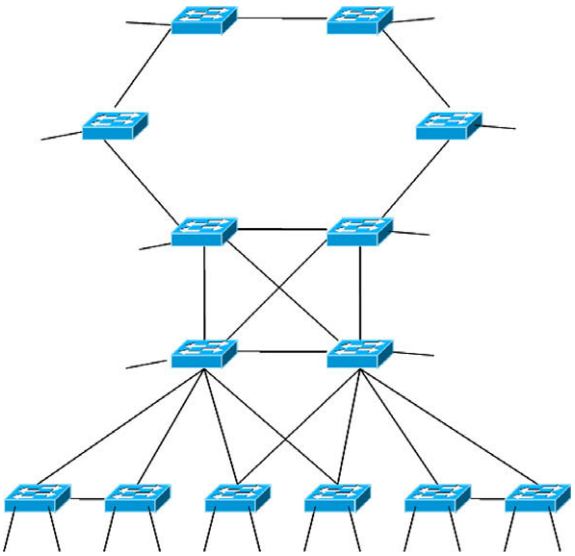


Ilustración 27. Ejemplo de topología de interconexión metro Ethernet

Nuevamente, HURP mejora sustancialmente los resultados de U/D en giros prohibidos y rendimiento relativo. La mejora no es tan acusada frente a STP y U/D debido a que la propia arquitectura tiene de por sí cierta semejanza con un árbol. En el caso del camino medio resultante HURP da los mismos resultados que SP aunque U/D tampoco lo empeora significativamente.

3.5 Conclusiones

HURP es una protocolo de conmutadores autoconfigurable y escalable, basado en el protocolo RSTP, el cual es extendido para la asignación de direcciones jerárquicas a los puentes y que utiliza el protocolo *Up/Down* para evitar bucles de tramas. La complejidad computacional de los protocolos es baja $O(N.d^2)$, mucho menor que otras propuestas que requieren conocimiento global de la topología. Puede implementarse como arquitectura única o de forma combinada pero independiente, coexistiendo con puentes estándar 802.1D en el mismo dispositivo mediante el direccionamiento local jerárquico HLMAC separado de las direcciones globales MAC. Además, permite el reenvío rápido

de tramas por el árbol de expansión empleando el direccionamiento jerárquico mediante descodificación paso a paso de la dirección destino, sin necesidad de tablas de encaminamiento.

El mecanismo auxiliar de encaminamiento es simple y admite diversas métricas. El mecanismo de prevención de bucles de HURP es aplicable a redes Ethernet de cualquier velocidad y a redes de interconexión. Las prestaciones son similares a otros protocolos como TP y TBTP con mucha menor complejidad y consistentemente mejores que *Up/Down*, más cercana a conmutadores de camino más corto (*shortest path*) y no dependen de manera significativa con la elección del puente raíz, aunque son posibles heurísticos de optimización con precálculo y elección previa del puente raíz.

La influencia relativa del número de giros prohibidos en las prestaciones disminuye al aumentar el grado medio de la red porque el número de giros prohibidos por nodo es $d(d-1)/2$ siendo d el grado del nodo, aumentando cuadráticamente con d por lo que al aumentar el grado de algunos nodos se crean muchos caminos alternativos en una topología. Por ello, la prohibición de algunos giros afecta en menor medida a la longitud de los caminos.

4 Protocolos TRE (*Tree-based routing Ethernet*), Enrutamiento basado en árbol en Ethernet

En este capítulo se describen las contribuciones realizadas en encaminamiento Ethernet basado en un árbol de expansión aumentado con conocimiento topológico de la localización de los nodos de la red, gracias al uso de direcciones jerárquicas. Con ello se consigue poder utilizar rutas de menor coste atravesando enlaces ajenos al árbol de expansión (atajos) con garantía de reenvío sin bucles. Se detallan los distintos protocolos que se han desarrollado, dando lugar a la Arquitectura TRE. Tras la introducción del tema se describen, primero, los trabajos previos y complementarios al protocolo que desembocan en el primer protocolo de la familia, el protocolo TRE original, y las distintas variantes propuestas que mejoran, progresivamente, el rendimiento del mismo: TRE+ y D-TRE. El desarrollo e implementación de los protocolos D-TRE se ha realizado en el marco de un proyecto fin de carrera dirigido por el autor.

4.1 Introducción

Los protocolos basados en la aplicación de la arquitectura TRAIN (*Tree-based Routing Architecture for Irregular Network* [CT97]) a Ethernet, presentan una interesante alternativa al encaminamiento basado tanto en protocolos de tipo *shortest path* como RSTP/STP. En su origen, TRAIN definió una arquitectura de árbol de expansión en la que se habilitaba el uso de ramas transversales, no pertenecientes al árbol (enlaces cruzados o "atajos"), para mejorar el rendimiento de la red, mediante la asignación de identificadores jerárquicos a cada nodo. La arquitectura resultante se denominó TRAIN, y constituye el punto de partida a partir del cual se desarrolla la arquitectura TRE (*Tree-based Routing Ethernet*) que engloba los protocolos TRE [IGC+09], TRE+ [CIG+10] y D-TRE [Rio12], que se proponen en esta Tesis.

La arquitectura TRE, aplicación de TRAIN a Ethernet, se propone para solventar las ineficiencias más importantes de RSTP/STP, derivadas del bloqueo de todos los enlaces ajenos al propio árbol de expansión, lo que produce un rendimiento global muy bajo, caminos medios largos (con alto retardo) y acumulación de rutas en las zonas altas del árbol (nodos cercanos al raíz), en general, serios problemas de escalabilidad. Además, se mantiene el objetivo de evitar la presencia de bucles, punto fuerte de RSTP y, a la vez, debilidad de los protocolos de tipo *shortest path*. Los protocolos *shortest path* requieren un conocimiento completo de la topología para calcular las rutas óptimas usando el algoritmo Dijkstra lo que produce que su complejidad computacional sea elevada y también limita su escalabilidad.

Comparados con el estándar RSTP, los protocolos de la Arquitectura TRE ofrecen mayor rendimiento, computado como el número de flujos existente en los enlaces "cuello de botella". Comparados con protocolos de tipo *shortest path*, ofrecen mayor protección contra bucles a la vez que reducen la complejidad en su implementación. En este sentido, los protocolos TRE+ y D-TRE alcanzan valores de rendimiento, en términos de coste mínimo de rutas, muy cercanos a *shortest path*.

4.2 Antecedentes: TRAIN (*"Tree-based Routing Architecture for Irregular Networks")*

Los multiprocesadores consiguen un rendimiento elevado, gracias al uso de redes de conmutadores con alto ancho de banda y baja latencia, para la interconexión de sus procesadores. Actualmente, los clústeres de estaciones de trabajo suponen una aproximación rentable para constituir una plataforma de multiprocesadores en paralelo, en la que se pueden utilizar las LAN tradicionales para proporcionar tráfico inter-nodo. Sin embargo, su rendimiento no está maximizado debido al ancho de banda limitado. La mayoría de estas redes usan topologías regulares (malla, toroide o hipercubo), para conectar los conmutadores. Se han desarrollado muchos algoritmos de enrutamiento para este tipo de bucles.

Aunque es posible usar redes regulares para interconectar procesadores masivos en paralelo, el incremento del número de procesadores hace que el uso de estas redes no sea rentable debido a su problema de escalabilidad. En comparación, una red de interconexión con topologías irregulares logra un mejor comportamiento de clúster. Estas redes irregulares consisten en conmutadores mucho más pequeños y se puede mejorar el rendimiento de la comunicación añadiendo más conmutadores, sin que se vea afectado el rendimiento de la red de forma significativa.

El objetivo es diseñar un sistema de encaminamiento eficiente, que proporcione un rendimiento alto y esté libre de bucles. Existen varias arquitecturas de encaminamiento para redes irregulares que evitan la presencia de bucles basándose en la utilización de una tabla de reenvío hardware en el conmutador. El hándicap de estos sistemas de encaminamiento, es que el tamaño de la tabla crece proporcionalmente a medida que aumenta el tamaño de la red. Por lo tanto, la implementación del conmutador es relativamente compleja, además del aumento de la latencia que sufren los paquetes en el momento de ser reenviados por el conmutador.

TRAIN propone una arquitectura de encaminamiento escalable para redes irregulares sin la necesidad de utilizar una tabla de reenvío en el conmutador. Su sistema de encaminamiento está libre de bucles y es adaptable a las condiciones dinámicas del tráfico en la red. Al no existir tabla de enrutamiento en el conmutador, la decisión de encaminamiento puede ser realizada más rápidamente, proporcionando baja latencia a los paquetes que recorren la red. Utiliza *cut-through* como técnica de reenvío.

La arquitectura TRAIN está basada en un árbol de expansión. Su estructura principal la forman los enlaces del árbol, que constituyen una ruta única desde cualquier nodo origen hasta cualquier nodo destino. Además, TRAIN incorpora la característica de que los demás enlaces, los que no forman parte del árbol, pueden ser usados como "atajos" para reducir el número de saltos que un paquete emplea en llegar a su destino, mejorando significativamente el rendimiento de la red.

4.2.1 Algoritmo de encaminamiento

TRAIN define un sistema de encaminamiento similar a Autonet [Sch+91] en tanto que también se usa un conjunto de nodos de la red para construir el árbol. Un paquete parte del nodo origen y sigue, por defecto, el camino marcado por el árbol para alcanzar el nodo destino. Con TRAIN, sin embargo, se pueden utilizar enlaces no presentes en el árbol, "atajos", para entregar el paquete. La idea básica de TRAIN es que un paquete entrante en un conmutador, y que debe ser reenviado, utilizará algún atajo si éste le proporciona una ruta más corta al nodo destino que la ruta por defecto a través del árbol. Estos "atajos" ayudan a reducir el número de saltos que un paquete invertirá hasta llegar al destino y, además, proporcionan más ancho de banda efectivo, mitigando la congestión de tráfico en el área de la red cercana a la raíz del árbol.

Cuando un paquete llega a un conmutador, el módulo de decisión de enrutamiento del conmutador comprueba si existe un atajo hasta algún conmutador más cercano al nodo destino. Si existe, el paquete es encaminado por el atajo. Si no existe atajo, o está actualmente bloqueado (no dispone de memoria suficiente para almacenar un nuevo paquete), el paquete continúa la ruta por el árbol, camino de la raíz. Hay que destacar que un paquete puede utilizar atajos en más de una ocasión a lo largo de la ruta desde el origen hasta el destino.

La clave del éxito del encaminamiento en TRAIN es que el cálculo de distancia entre dos nodos es simple, realizándose mediante un algoritmo hardware, sin necesitar una tabla de encaminamiento en el conmutador.

La Ilustración 28 muestra un ejemplo de red que utiliza la arquitectura TRAIN. El número que aparece junto a cada nodo es su ID, empleada en el cálculo de distancias. El método de etiquetado es el siguiente: La raíz se etiqueta como '00…0' (el árbol no tiene que ser necesariamente un árbol binario). Los hijos de la raíz se etiquetan '10…0', '20…0', y así sucesivamente. De esta manera, desde el nodo '121' hasta el destino '220', la ruta por defecto a través del árbol requiere cinco saltos, como se observa en la Ilustración 28. Sin embargo, cuando el paquete llega al nodo '120', se descubre que existe un atajo a través el nodo '200' que proporciona una ruta más corta. Por lo tanto, siguiendo este

atajo, la distancia desde el nodo ‘121’ hasta el nodo ‘220’ se reduce, en este caso, de cinco a tres saltos.

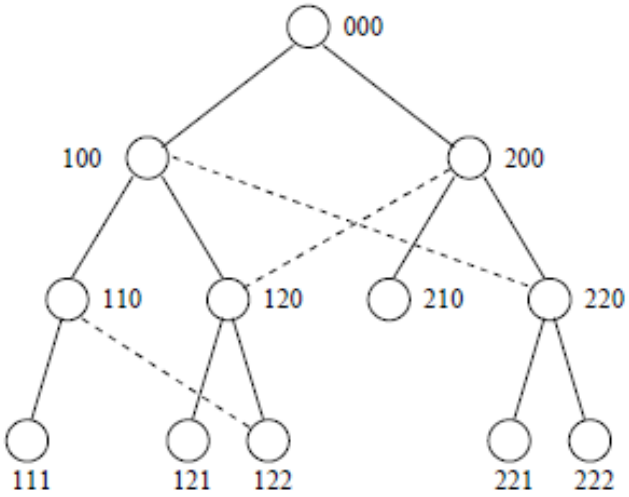


Ilustración 28. Ejemplo de direccionamiento en TRAIN

Es preciso destacar que para que funcione el módulo del conmutador que implementa la decisión de encaminamiento, es decir, que un conmutador descubra atajos que proporcionan rutas más cortas a través de algún vecino, es necesario que el conmutador almacene las IDs de los nodos vecinos. El conmutador consigue esta información durante la fase de inicialización.

4.2.2 Cálculo de la distancia entre dos nodos

El cálculo de distancias entre dos nodos es un proceso simple pero crítico en TRAIN para conseguir un rendimiento elevado. La solución a este problema se implementa mediante un algoritmo en dos pasos:

1. Se elimina la cadena que forma el prefijo común de las IDs de los dos nodos.

2. La distancia entre los dos nodos es exactamente igual al número de dígitos distintos de cero que quedan en las dos cadenas.

Este algoritmo de cálculo de distancias se aplica, como se verá más adelante, en los protocolos pertenecientes a la familia TRE, (TRE, TRE+ y D-TRE, protocolos diseñados en este trabajo).

En la Ilustración 29 se muestra un ejemplo de cálculo de distancias entre varios nodos pertenecientes a la red de la Ilustración 28. Para el caso de la izquierda, no hay dígitos que formen un prefijo común y por lo tanto la distancia del nodo ‘100’ al nodo ‘220’ es tres, ya que hay tres dígitos distintos de cero. Para el caso de la derecha, la distancia desde el nodo ‘200’ al nodo ‘220’ es uno, porque el prefijo común “2” es eliminado de ambas cadenas y, posteriormente, solo queda un dígito distinto de “0”.

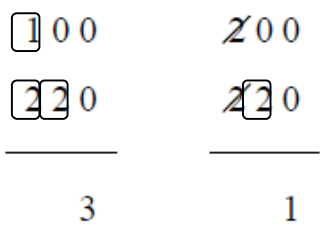


Ilustración 29. Cálculo de distancias entre dos nodos en la arquitectura TRAIN

4.2.3 Algoritmo de reenvío

El algoritmo global de reenvío en TRAIN consta de cuatro fases sucesivas:

1. Para cada paquete entrante en un conmutador, se calcula la distancia desde el nodo actual hasta el nodo destino, comparándose la ID del nodo destino del paquete con la ID almacenada de los nodos vecinos, mediante el algoritmo descrito en el párrafo anterior.

2. Si algún atajo, que no se encuentre bloqueado en ese momento, proporciona una ruta más corta que la ruta del árbol, se encamina el paquete hacia el nodo vecino unido al nodo actual a través de dicho atajo.

3. Si no existen atajos que proporcionen rutas más cortas que la del árbol o están bloqueados, el conmutador encamina el paquete hasta el nodo padre a través del árbol.

4. Si todos los atajos y el enlace ascendente del árbol están bloqueados en el ciclo actual, se espera al siguiente ciclo y se repiten los pasos descritos desde la Fase 2.

4.2.4 Reenvío sin bucles

El encaminamiento TRAIN consigue evitar la formación de bucles. La técnica de reenvío empleada en TRAIN es *cut-through*, la cual forma parte también del mecanismo de prevención de bucles. *Cut-through* es una técnica en la que el paquete puede comenzar a reenviarse al siguiente nodo sin que se haya recibido por completo, una vez que se ha recibido la cabecera y se ha decidido el puerto de salida. Esta técnica requiere que el siguiente conmutador tenga suficiente espacio en buffer para almacenar el paquete. De esta manera se evita que se pueda producir bloqueo en el conmutador destino en el comienzo del envío (cabecera del paquete) y el paquete siga enviándose, ya que el paquete permanecerá en el conmutador destino mientras dure el bloqueo. El uso de *cut-through* evita este tipo de bloqueo, ayudando a reducir la congestión de la red.

El mecanismo de prevención de bucles en TRAIN es sencillo y se basa en que los conmutadores buscan “atajos” alternativos a los enlaces ascendentes, sin embargo, una vez que se alcanza la rama del destino, se sigue la ruta descendente del árbol sin desviación posible (sin buscar atajos, ya que el camino es el mejor posible puesto que forma parte de una rama del árbol de expansión).

En la Ilustración 30, los paquetes recibidos en el nodo B (procedentes del nodo A) no utilizarán nunca el atajo que conecta el nodo B con el nodo D porque corresponde a un enlace descendente, es decir, no se buscarán atajos y, por lo tanto, el paquete no irá a otra rama del árbol. Por otro lado, los paquetes recibidos en el nodo B procedentes del nodo C pueden elegir utilizar el atajo que une B con D o ir hacia el nodo A, dependiendo de su destino y del cálculo de distancias.

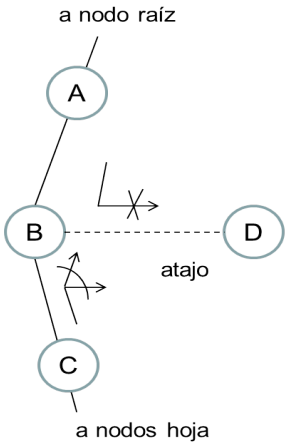


Ilustración 30. Prevención de bucles en TRAIN

En TRAIN, un atajo que proporciona un camino más corto hasta un destino dado tiene mayor prioridad que el enlace del árbol hacia la raíz, a no ser que dicho atajo esté bloqueado. En la Ilustración 30 los paquetes que se reciban en el nodo B desde C, o desde cualquier otro atajo, seguirán el enlace del árbol hasta A si el atajo hasta D está bloqueado, aunque proporcione una ruta más corta.

4.3 El protocolo TRE (*"Tree-based Routing Ethernet"*)

TRE [IGC+09] implementa una combinación de funciones que permiten aplicar el principio básico de TRAIN a Ethernet. Su objetivo es conocer si hay algún vecino que, ajeno al árbol, proporcione un camino con menor número de saltos que a través del puerto raíz, logrando, como ventaja añadida, menor ocupación en los enlaces superiores del árbol.

Para construir el árbol y asignar direcciones jerárquicas a cada nodo, TRE usa una extensión de RSTP, tal como se define en HURP [Iba+08b] y se detalla a continuación. El cálculo de distancias entre dos nodos se realiza como se ha explicado en el apartado dedicado a TRAIN, al ser un protocolo que aplica su arquitectura a Ethernet.

4.3.1 Direccionamiento en TRE

TRE requiere que se asigne a cada conmutador una dirección MAC jerárquica y local, HLMAC (*Hierarchical Local MAC*). Tal y como se explica en el capítulo anterior, cada dirección MAC está formada por seis octetos. Para diferenciar este direccionamiento como *local*, el bit "1" del octeto "0" se establece a '1'. Los 46 bits disponibles para direccionamiento se utilizan para codificar, por defecto, seis niveles jerárquicos diferentes, con seis bits disponibles para el primer nivel del árbol y ocho bits para cada uno de los otros cinco niveles.

La dirección HLMAC del conmutador se expresa en la notación con puntos "a.b.c…", igual que los identificadores de los puertos designados (ID), "a.b.c…", formando el camino descendente desde la raíz del árbol hasta el conmutador cuya dirección está siendo asignada. Se muestra un ejemplo en la Tabla 16 para el conmutador con dirección '5.22.9.3.4.0' ó '5.22.9.3.4' en forma abreviada.

Tabla 16. Ejemplo de codificación de una dirección HLMAC

Nº octeto	1	2	3	4	5	6
Bits	01000101	00010110	00001001	00000011	00000100	00000000
Valor decimal	5	22	9	3	4	0

Para construir el árbol y asignar direcciones jerárquicas, TRE utiliza la misma versión de RSTP diseñada para HURP. Una vez que la raíz ha sido elegida, de acuerdo con el estándar RSTP, se le asigna la dirección HLMAC '0.0.0.0.0.0', envía su BPDU y comienza el proceso estándar de construcción del árbol. Una vez establecido el origen del árbol, los conmutadores reciben las BPDUs de la raíz y son configurados con la dirección resultante de sustituir el primer "0" existente en la dirección del conmutador raíz por el número del puerto designado incluido en la BPDU. Posteriormente reenvían la trama para que todos los nodos realicen este proceso. En la Ilustración 31, el conmutador '1.18.0.0.0.0' ha configurado su dirección después de recibir una BPDU enviada desde el conmutador con HLMAC '1.0.0.0.0.0' a través de su puerto designado número "18". TRE no requiere para la construcción del árbol ningún envío de tramas de control adicionales a RSTP.

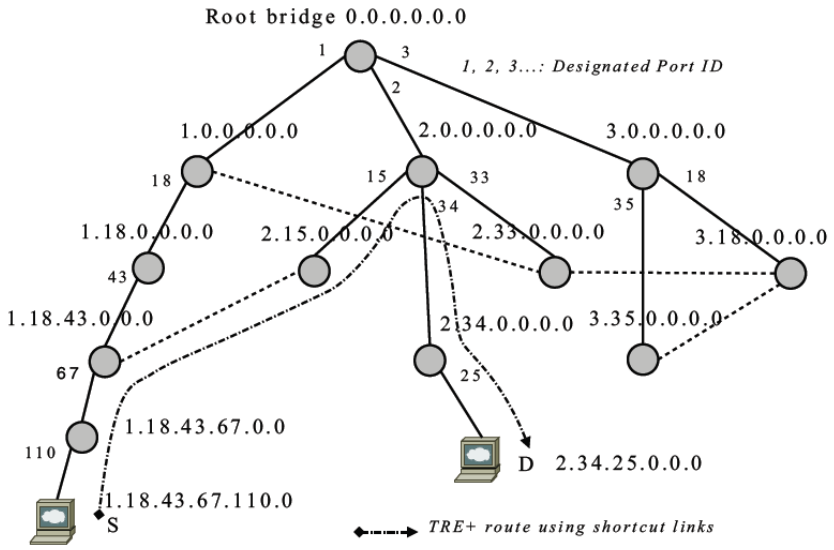


Ilustración 31. Direcciones jerárquicas y ejemplo de funcionamiento de TRE

4.3.2 Reenvío de tramas

El reenvío (*forwarding*) unicast se desarrolla en cada conmutador como se explica a continuación y se recoge en el diagrama de flujo de la Ilustración 32.

En primer lugar, el conmutador comprueba si el destino pertenece a su propia rama del árbol. Por ejemplo, la dirección '2.34.25.0.0.0' pertenece a la misma rama que '2.0.0.0.0.0' porque tienen un prefijo común, "2" (la parte distinta de cero de la segunda dirección está incluida en la primera). Si ambos puentes están localizados en la misma rama, se usa el camino establecido por STP/RSTP.

El puerto por el que se retransmite un paquete a un conmutador destino situado en la misma rama, se determina comparando la dirección HLMAC del conmutador que retransmite la trama y la del destino. Si la dirección destino está contenida en la dirección del conmutador actual, se retransmite el paquete por el puerto raíz de dicho conmutador, es decir, la trama debe ascender (por el árbol hacia la raíz). Por otro lado, si la dirección del conmutador actual está contenida en la dirección destino, el destino está situado por debajo del conmutador en la misma rama, es decir, la trama debe descender. En este caso, el número de puerto para el reenvío se obtiene de la dirección destino, siendo el primer elemento no común entre la dirección del conmutador y el destino. Por ejemplo, si el conmutador '2.0.0.0.0.0' recibe una trama dirigida a '2.34.25.0.0.0', el camino es descendente, porque la dirección del destino contiene a la dirección del conmutador actual y el siguiente elemento tras la parte común ("2"), "34", identifica el número del puerto de reenvío. Es importante destacar que todos los nodos pertenecientes a la misma rama están conectados por el camino más corto, porque los caminos que son parte de la ruta óptima son a su vez caminos óptimos.

Si el destino no se encuentra en la misma rama que el conmutador que reenvía la trama, se considera el uso de atajos, siempre que se tenga la seguridad de que la distancia a través del atajo es menor que a través del árbol. Para tomar esta decisión, la distancia entre dos HLMAC se calcula como se ha explicado para TRAIN en el apartado 4.2.2. Resumiendo, el prefijo común de ambas direcciones es identificado, si existe, y se elimina del cómputo. La distancia se define entonces como la suma de los elementos no nulos que queden en ambas direcciones. Por ejemplo, la distancia entre '2.15.0.0.0.0' y '2.34.25.0.0.0', siguiendo el árbol de expansión, es tres, porque el primer elemento en las direcciones, "2", es el mismo en ambas, y después de descartar este identificador común y los campos a cero, quedan tres elementos ("15", "34" y "25"), que indican los tres saltos necesarios para ir de una a otra (y por tanto una distancia de valor tres).

Cuando un conmutador 'S' recibe una trama con destino a un nodo 'D', localizado en una rama diferente, 'S' calcula la distancia, siguiendo el árbol de expansión, entre el siguiente vecino superior en la rama del árbol (su padre) y 'D'. Posteriormente, considera las distancias de cada uno de sus vecinos 'i1', 'i2', etc. hasta 'D'. Si la distancia al destino a través de algún vecino conectado es menor que la obtenida a través del árbol, se selecciona el atajo (teniendo en cuenta que hay que añadir un salto a la ruta del vecino al nodo destino 'D', que es la distancia del nodo 'S' a su vecino 'i'). La Ilustración 31 muestra la operación de reenvío del algoritmo, mostrando con una línea discontinua la ruta seguida por una trama desde un nodo origen 'S', con HLMAC '1.18.43.67.110.0' hasta un nodo destino 'D', con dirección HLMAC '2.34.25.0.0.0'. En el conmutador '1.18.43.0.0.0', la distancia al destino a través del atajo es menor que a través del árbol, por lo tanto, la trama es encaminada por el atajo.

El reenvío unicast en TRE puede seguir el árbol, libre de bucles, o seguir atajos, los cuáles son elegidos cuando la distancia al destino es estrictamente más corta que a través del árbol. Después de cada salto, bien sea a través del árbol o del atajo, la trama está al menos un salto más cerca del destino. Por lo tanto, la trama llegará al destino en un número de saltos finito.

Tanto el envío *multicast* como la difusión se realizan a través del árbol de expansión, como ocurre en la Ethernet clásica.

En la Ilustración 32 se muestra el diagrama de flujo del procedimiento de reenvío de tramas de datos desde un nodo 'S' hasta un nodo 'D'.

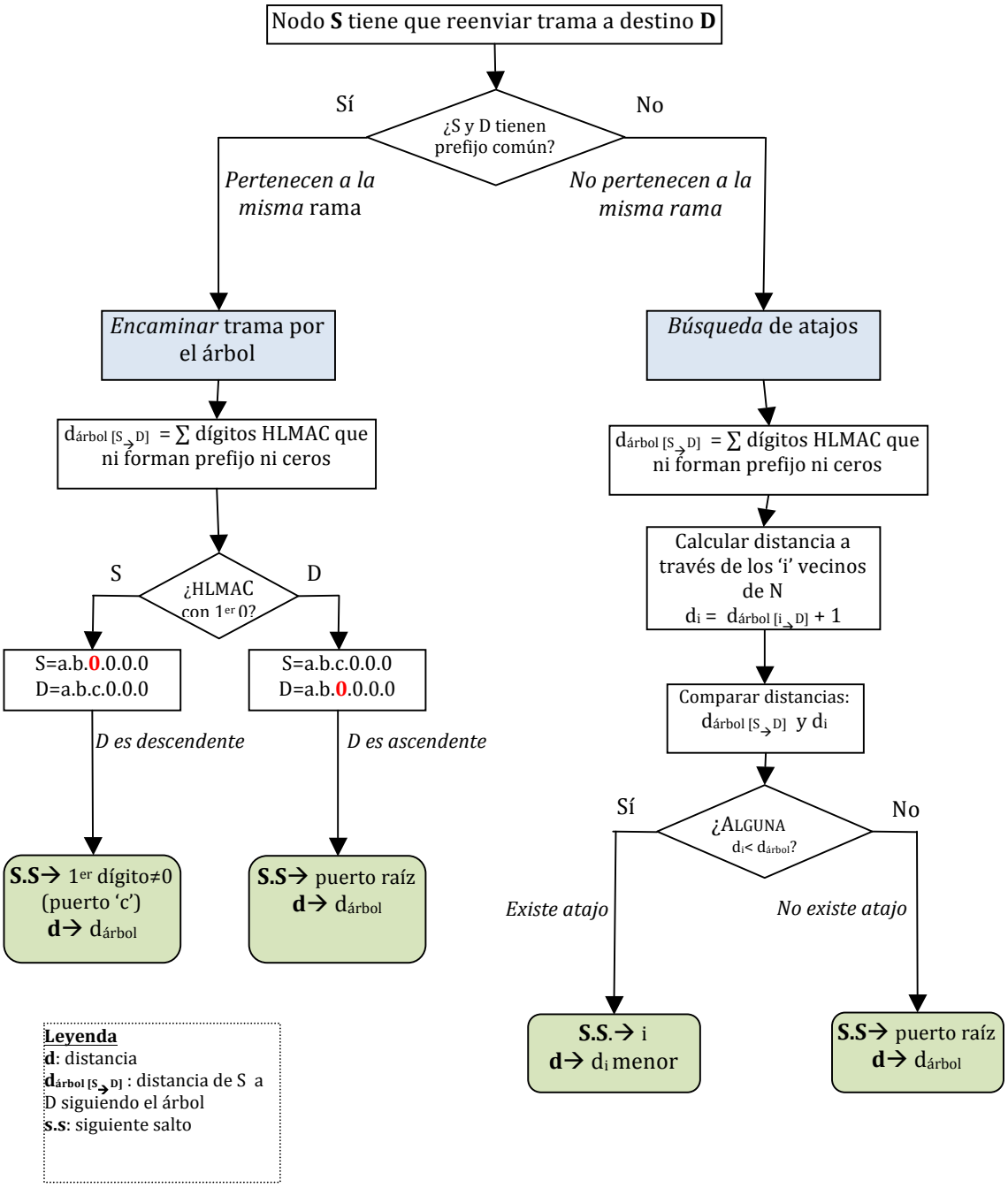


Ilustración 32. Procedimiento de Reenvío de tramas de datos en TRE

4.3.3 Análisis de estabilidad

Podemos afirmar que TRE está libre de bucles en estado estacionario, es decir, cuando las direcciones han sido configuradas de manera definitiva mediante las extensiones indicadas al protocolo RSTP. En caso de que se produzca un cambio en la topología, por ejemplo cuando falla un enlace, falla un conmutador o durante la fase de inicialización de alguno de ellos, TRE se apoya en los mecanismos de RSTP para reconfigurar el árbol y reasignar las direcciones HLMAC que se necesite.

Cuando el conmutador recibe una notificación de que la topología ha cambiado, desactiva la dirección HLMAC asignada y detiene el reenvío de tramas, hasta que se reconfigure el árbol de expansión y las nuevas direcciones HLMAC. Por lo tanto, se evitan los bucles de transición.

El impacto sobre el rendimiento del protocolo es similar a RSTP ya que TRE descansa en el propio algoritmo que restablece el estado de un puerto a *forwarding* para validar la nueva dirección jerárquica.

4.3.4 Análisis de prestaciones

El rendimiento de TRE se ha evaluado mediante dos estadísticos básicos: midiendo la longitud media del camino, calculada como el número medio de saltos desde cada nodo origen hasta los posibles destinos, y el *throughput* relativo, calculado como el número de flujos unitarios que transcurren por el enlace más ocupado de la red, tomando como referencia el valor obtenido mediante un protocolo de tipo *shortest path*.

El *throughput* relativo se calcula dividiendo el número de flujos que pasan por el enlace más ocupado (cuello de botella de la red), entre el obtenido para *Shortest Path (SP)*. Hay que tener en cuenta que *SP* puede no ser óptimo, en este aspecto, para alguna topología concreta, ya que obtener el máximo rendimiento requeriría solucionar un problema de optimización de la distribución de carga. En cambio, *SP* siempre va a ser óptimo en el número de saltos de una ruta, ya que los conmutadores que implementan este protocolo disponen del conocimiento topológico de toda la red.

El modelo de tráfico empleado para el análisis consiste en el establecimiento de un flujo de datos unitario entre cada par de nodos de la red (N-1 flujos por nodo), por tanto, se considera tráfico uniformemente distribuido. Las topologías de red evaluadas también responden a modelos aleatorios de generación de nodos y enlaces entre nodos según distintos modelos.

Para evaluar el rendimiento de TRE se han simulado distintos tipos de topologías de redes irregulares generadas mediante la herramienta BRITE [BRITE], con distintos grados medios (4, 6 y 8) y tamaños (número de nodos total en la red) en un simulador construido en Python [PYTHON]. Los modelos generados son de escala libre, *scale-free*, (modelos Barabasi-Albert [AB02][BA99][Bar02][FFF99] que siguen una distribución exponencial) y aleatorios (modelos Waxman). Para eliminar la dependencia en una raíz concreta elegida, se realizan distintas iteraciones por topología, estableciendo una raíz distinta en cada iteración. Los resultados que se muestran son las medias de las iteraciones por cada topología, y las medias de las topologías de cada tipo concreto (modelo, grado medio y tamaño).

La Tabla 17 y la Tabla 18 muestran los valores de rendimiento comparado de TRE, SP y STP en cuanto a la longitud del camino medio en la red, para topologías aleatorias libres de escala (modelo Barabasi-Albert) para distintos tamaños (64, 128 y 256 nodos) y grado medio (4, 6 y 8) de la red.

Tabla 17. Comparativa de longitud de camino medio en redes aleatorias de escala libre (modelo Barabasi-Albert)

Barabasi-Albert		Longitud del camino medio		
Nº Nodos	Grado medio	SP	TRE	STP
64	4	2,8	3,1	3,8
	6	2,4	2,8	2,5
	8	2,2	3,6	3,4
128	4	3,1	3,5	4,6
	6	2,7	3,2	4,1
	8	2,4	2,9	3,8
256	4	3,5	4,2	5,2
	6	3,0	3,5	4,5
	8	2,7	3,3	4,3

Tabla 18. Comparativa de rendimiento relativo (*throughput* frente a Shortest Path) en redes aleatorias de escala libre (modelo Barabasi-Albert)

Barabasi-Albert		Throughput (%) SP		Ratio Th.
Nº Nodos	Grado medio	TRE	STP	TRE/STP
64	4	58,4	25,0	2,36
	6	46,9	19,2	2,44
	8	36,2	10,2	3,55
128	4	46,9	19,2	2,44
	6	32,3	10,1	3,2
	8	28,7	7,4	3,88
256	4	36,5	15,4	2,37
	6	29,8	9,2	3,24
	8	22,5	5,6	4,02

Las tablas Tabla 19 y Tabla 20 muestran los valores de rendimiento relativo frente a *Shortest Path* de TRE y STP en topologías aleatorias (modelo Waxman) para distintos tamaños (64, 128 y 256 nodos) y grado medio (4, 6 y 8) de la red.

Los resultados obtenidos indican que el ratio de rendimiento relativo entre TRE y STP se incrementa cuanto mayor es el grado medio, como consecuencia del mayor número de atajos, y muestra poca dependencia en el tamaño de la red. Sin embargo, el incremento con el grado medio no se respeta con SP, debido a las limitaciones de TRE: solo se pueden usar atajos a la rama en la que el destino está localizado, y cada conmutador toma la decisión de elegir un atajo en base a sus vecinos, aunque se pueda tomar un mejor atajo posteriormente en la rama.

Tabla 19. Comparativa de longitud de camino medio en redes aleatorias (modelo Waxman)

Waxman		Longitud del camino medio		
Nº Nodos	Grado medio	SP	TRE	STP
64	4	2,99	3,57	4,42
	6	2,5	3,04	3,9
	8	2,22	2,74	3,63
128	4	3,52	4,47	5,66
	6	2,86	3,71	4,89
	8	2,56	3,34	4,5
256	4	4,02	5,23	6,28
	6	3,26	4,36	5,36
	8	2,89	3,87	4,85

Tabla 20. Comparativa de rendimiento relativo (*throughput* frente a Shortest Path) en redes aleatorias (modelo Waxman)

Waxman		Throughput (%) SP		Ratio Th.
Nº Nodos	Grado medio	TRE	STP	TRE/STP
64	4	38,4	18,6	2,06
	6	31,4	12,2	2,57
	8	27,1	10,0	2,71
128	4	28,5	12,9	2,21
	6	22,2	7,44	2,99
	8	19,5	5,6	3,48
256	4	18,5	9,02	2,05
	6	12,9	5,13	2,51
	8	12,4	4,38	2,82

El ratio de rendimiento TRE/STP es ligeramente menor en topologías aleatorias puras porque en distribuciones sin escala existen nodos con alto grado (con muchos vecinos) que actúan como concentradores/distribuidores, a modo de *hubs* de la red, cuyos enlaces son susceptibles de utilizarse como atajos para alcanzar distintas partes de la red sin necesidad de recorrer todo el camino hasta la raíz. Aunque SP ofrece mejores resultados, TRE es una buena alternativa, debido a su baja complejidad y a su mecanismo de reenvío libre de bucles.

4.4 El protocolo TRE+ ("*Extended* TRE")

El objetivo de TRE+ [CIG+10], que supone además su principal característica diferenciadora respecto a TRE, consiste en la extensión del conocimiento que cada nodo posee de la topología hasta dos saltos de distancia, en lugar de uno, con el fin de revelar nuevos atajos a la rama destino. TRE+ obtiene aún mejor rendimiento que TRE respecto a STP, con resultados muy próximos a *Shortest Path* en algunas topologías.

Esta extensión se argumenta con el fin de obtener más información en cada conmutador y así incrementar el número de caminos alternativos no pertenecientes al árbol, "atajos", por los que retransmitir una trama, logrando mejores prestaciones en la red. Para conseguirlo, el conmutador TRE+ debe informar, a cada uno de sus vecinos, de los vecinos que posee. Este intercambio de rutas, al estilo de un vector de distancias clásico, se propaga únicamente a nodos vecinos (nodos situados a un salto de distancia) para prevenir la aparición de bucles de transición resultantes de cuentas hasta infinito.

4.4.1 Direccionamiento

TRE+ se fundamenta en la misma arquitectura que TRE, por lo tanto, parte de la topología en árbol, proporcionada por un protocolo estándar como RSTP, y comparte el modelo de direccionamiento, la asignación de direcciones jerárquicas HLMAC y el procedimiento de construcción del árbol detallados en el apartado anterior para TRE. El cálculo de distancias entre dos nodos TRE+ es el mismo que para TRE, es decir, tal cual se describe en el apartado 4.2.2 para la arquitectura TRAIN.

4.4.2 Descubrimiento de nodos a dos saltos

TRE+ requiere un nuevo mecanismo de detección de conmutadores localizados a dos saltos de distancia (nodos vecinos de sus vecinos) y así aprender sus direcciones HLMAC para considerarlas en las reglas de reenvío.

El procedimiento para descubrir conmutadores situados a dos saltos es similar al intercambio de vectores de distancia, aunque en este caso, el alcance de los intercambios está limitado a un salto. En otras palabras, cada conmutador notifica periódicamente a cada uno de sus vecinos las direcciones HLMAC de todos los conmutadores a los que está directamente conectado, es decir, a una distancia de un salto.

En la figura siguiente el nodo N notifica al nodo A, situado a un salto de distancia, cuáles sus vecinos directos, B y C, es decir, los que están igualmente a un salto. Realiza la misma operación para el resto de sus nodos vecinos. Al nodo B le informa de que A y C son también vecinos de N y, del mismo modo, al nodo C le informa de la alcanzabilidad de A y B, desde N, en un salto. El nodo A elabora su tabla de reenvío en base a los mensajes "vector-distancia" recibidos de sus vecinos, en el ejemplo solo el nodo N.

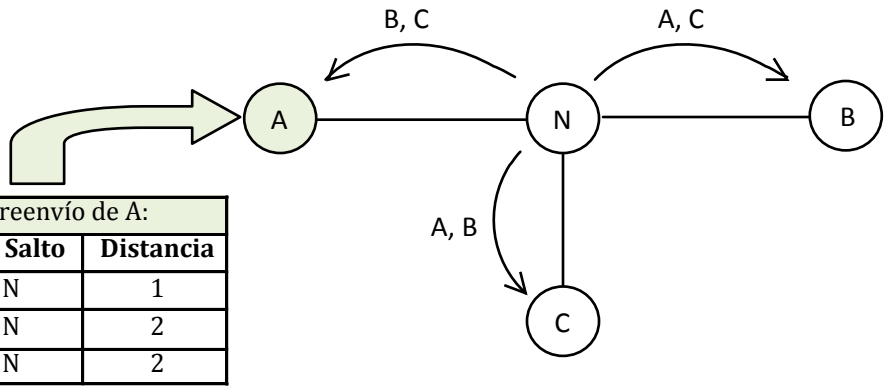


Tabla de reenvío de A:		
Destino	**Sig. Salto**	**Distancia**
N	N	1
B	N	2
C	N	2

Ilustración 33. Intercambio de información de vecindad en TRE+ y tablas resultantes

Cuando un conmutador pierde comunicación con uno de sus vecinos, inmediatamente notifica distancia dos, considerada como distancia infinita en esta implementación, ya que la distancia que notifica de sus vecinos es siempre uno cuando el enlace es estable.

4.4.3 Reenvío de tramas

Cada conmutador TRE+ tiene asignada una dirección HLMAC, de la misma manera que en TRE (ver apartado 4.3.1). Cada trama recibida que no tenga como destino una dirección HLMAC de la misma rama (mismo prefijo), se enviará por la rama ascendente del árbol, a menos que la distancia calculada a través de un atajo sea estrictamente menor. Por lo tanto, el conmutador debe primero calcular la distancia al destino a través del árbol (de la misma manera que en TRE, según se ha explicado en el apartado 4.2.2 para la arquitectura TRAIN), posteriormente a través de posibles atajos y, por último, elegir el puerto que proporcione la ruta más corta. Para calcular la distancia a través de un atajo ofertado por un nodo I, el conmutador S debe obtener la distancia desde I al destino, usando la ruta por defecto del árbol y, después, añadir su propia distancia a I, sean uno o dos saltos. El procedimiento de TRE+ es el que realiza TRE (diagrama de flujo de la Ilustración 32), pero con una modificación, en la búsqueda de atajos, en lugar de añadir siempre un salto a la distancia de los vecinos al nodo destino, se añade la distancia que aparece en la tabla de reenvío, que será un salto si el atajo lo proporciona un vecino, o dos saltos si lo proporciona un vecino de vecino.

Un ejemplo del proceso de reenvío de TRE+ se presenta en la Ilustración 34. El nodo S envía una trama al nodo D. El nodo S retransmite la trama hacia arriba en el árbol a '1.7.0.0.0.0' (resumido en '1.7') a través de su puerto raíz. La distancia del nodo '1.7' al destino D, usando estrictamente el árbol, son seis saltos. A continuación '1.7' comprueba la posibilidad de utilizar atajos que proporcionen una ruta menor. Los nodos '14', '8' y '8.6' se encuentran a distancia máxima de dos saltos y no pertenecen a la misma rama que '1.7', por lo que son considerados para el cómputo. La mínima distancia calculada por '1.7', teniendo en cuenta esos tres nodos, es cinco, obtenida a través del nodo '8', siendo su ruta teórica [1.7 – 14 – 8 – 8.9 – 8.9.1 – 8.9.1.25]. Por lo tanto, al obtenerse una distancia menor que la calculada a través del árbol, la trama es reenviada a través del nodo '14'.

Sin embargo, al llegar al nodo '14' se vuelve a ejecutar el algoritmo y se evalúan los atajos proporcionados por los nodos '1', '8', '8.9', '8.6' y '8.9.1'. El nodo '14' descubre entonces a través de

'8.6' el nodo '8.9.1', directamente conectado con el destino, por lo que la trama será reenviada a '8.6' y seguirá la ruta [14 – 8.6 – 8.9.1 – 8.9.1.25].

En este ejemplo, la ruta obtenida desde S hasta D a través del árbol se efectúa en siete saltos, mientras que con el protocolo TRE se llevaría a cabo en seis saltos. La ruta se optimiza con el protocolo TRE+, consiguiendo alcanzar el destino en solo cinco saltos.

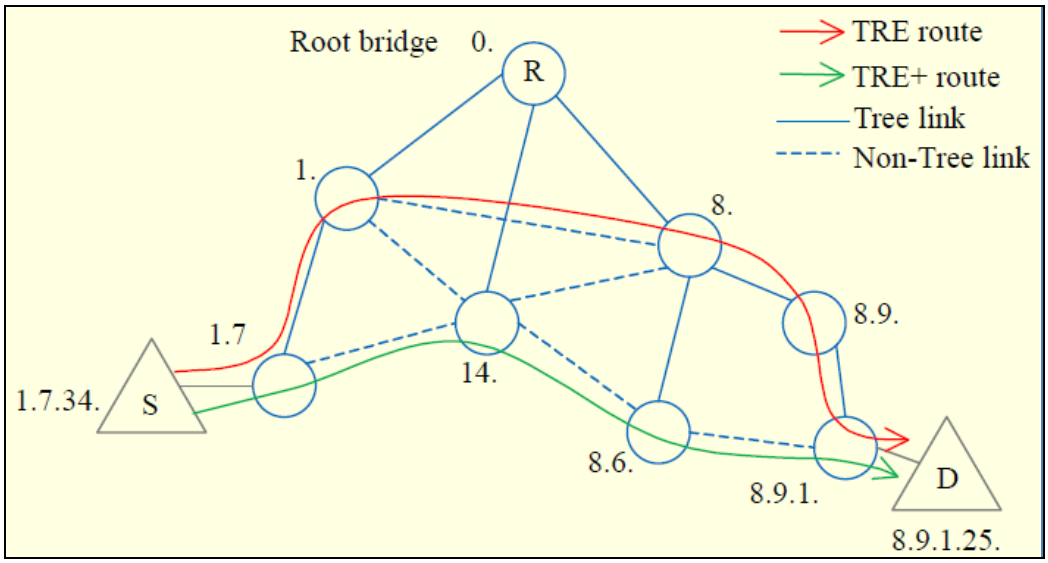


Ilustración 34. Ejemplo de funcionamiento de TRE+ y comparación con TRE

4.4.4 Análisis de estabilidad

TRE+ es estable cuando las direcciones se han asignado de forma permanente. El camino definido por el atajo está libre de bucles en estado estacionario porque es el resultado de un intercambio de mensajes tipo vector-distancia simple. La combinación de los caminos del árbol y los atajos está también libre de bucles porque un atajo es elegido exclusivamente cuando la distancia al destino es estrictamente más pequeña que a través del árbol. Después de cada salto, la distancia al destino es más baja que la estimada en el conmutador anterior. Por lo tanto, una trama debe llegar al destino en un número finito de saltos.

Para analizar el comportamiento de la topología cuando sucede un cambio, se consideran por separado fallos que afectan a los atajos y fallos que afectan al árbol.

Si se utilizan atajos en condiciones de estabilidad, nunca se podrían producir bucles circulares que incluyan tres o más conmutadores, ya que la máxima distancia que se considera a un destino es dos. Por lo tanto, solo podrían ocurrir bucles si dos conmutadores incorrectamente asumen que el otro está conectado directamente con el destino y se produce un fallo. Considerando dos conmutadores S1 y S2, directamente conectados con D, esta situación ocurre si ambos envían un aviso con distancia uno a D, justo antes de que D falle (o lo hagan ambos enlaces S1-D y S2-D). S1 inmediatamente envía un aviso a S2 informando distancia dos hasta D, que equivale a distancia infinita, pero entonces recibe un aviso desactualizado de S1 con distancia uno. Existe un breve periodo de tiempo, hasta que llega el nuevo aviso actualizado, en el que se puede presentar inestabilidad en estos enlaces. Para prevenir esta situación, los avisos de abandono enviados por el conmutador S1 a S2, referentes a D, incluyen un número único aleatorio y fuerzan a S2 a enviar un nuevo aviso incluyendo este número, si S2 sigue conectado directamente a D. S1 no valida la

información que recibe desde S2 sobre D hasta que el número es devuelto. Mientras tanto, S2 no es considerado como un atajo válido a D.

Si el fallo afecta al árbol, TRE+ se basa en los mensajes RSTP para reconfigurar el árbol de expansión y reasignar las direcciones HLMAC. Cada conmutador que recibe estos mensajes, detiene inmediatamente el reenvío hasta que se reconfiguren las HLMAC. Igualmente se elimina la información de vecindad de cada conmutador.

4.4.5 Análisis de prestaciones

El rendimiento de TRE+ se ha evaluado, de la misma manera que en TRE, midiendo la longitud media del camino, computada como número de saltos de cada nodo origen hasta los posibles destinos, y el *throughput* relativo, computado como número de flujos unitarios que transcurren por el enlace más ocupado de la red, relativo a un protocolo de tipo *shortest path*.

El modelo de tráfico empleado para el análisis también es el utilizado en el análisis del rendimiento de TRE, se modelan flujos unitarios desde cada nodo a todos los demás nodos. De esta manera es posible comparar el rendimiento de TRE+ con TRE, STP y *Shortest Path (SP)*.

Para evaluar el rendimiento de TRE+ se han simulado distintos tipos de topologías de redes irregulares generadas con BRITE [BRITE], con distintos grados medios (4, 6 y 8) y tamaño (número de nodos en la red) mediante un simulador en Python [PYTHON]. Los modelos generados son de escala libre, "*scale-free*", (modelos Barabasi-Albert que siguen una distribución exponencial) y aleatorios (modelos Waxman). Para eliminar la dependencia en una raíz concreta elegida, se realizan distintas iteraciones por topología, estableciendo una raíz distinta en cada iteración. Los resultados que muestran corresponden con las medias de las iteraciones realizadas para cada topología y las medias de las distintas topologías.

El rendimiento de las topologías de escala libre (Barabasi-Albert) y de las topologías aleatorias (Waxman) se muestran en la Tabla 21 y Tabla 22, respectivamente.

Tabla 21. Rendimiento de TRE+ en topologías sin escala (Barabasi-Albert)

Topología		**Longitud media del camino**				**Throughput (% de SP)**			**Ratio Th.**	
Nodos	Grado medio	SP	TRE+	TRE	STP	TRE+	TRE	STP	TRE+/ TRE	TRE+/ STP
64	4	2.8	2.8	3.1	3.8	91.9	58.4	25.0	1.6	2.3
	6	2.4	2.4	2.8	3.6	96.2	41.0	13.6	2.3	3.0
	8	2.2	2.2	2.5	3.3	97.2	36.2	10.2	2.7	3.5
128	4	3.1	3.2	3.5	4.6	89.5	46.9	19.2	1.9	2.4
	6	2.7	2.8	3.2	4.1	84.6	32.3	10.1	2.6	3.2
	8	2.4	2.5	2.9	3.8	91.1	28.7	7.4	3.2	3.9
256	4	3.5	3.7	4.2	5.2	75.5	36.5	15.4	2.1	2.4
	6	3.0	3.1	3.5	4.5	80.9	29.8	9.2	2.7	3.2
	8	2.7	2.8	3.3	4.3	82.7	22.5	5.6	3.7	4.0

Tabla 22. Rendimiento de TRE+ en topologías aleatorias (Waxman)

Topología		Longitud media del camino				Throughput (% de SP)			Ratio Th.	
Nodos	Grado medio	SP	TRE+	TRE	STP	TRE+	TRE	STP	TRE+/ TRE	TRE+/ STP
64	4	2.99	3.20	3.57	4.42	76.2	38.4	18.6	2.0	4.1
	6	2.5	2.60	3.04	3.9	86.0	31.4	12.2	2.7	7.0
	8	2.22	2.30	2.74	3.63	92.0	27.1	10.0	3.4	9.2
128	4	3.52	3.90	4.47	5.66	60.8	28.5	12.9	2.2	4.6
	6	2.86	3.10	3.71	4.89	65.5	22.2	7.44	3.1	8.1
	8	2.56	2.70	3.34	4.5	70.0	19.5	5.6	3.9	11.1
256	4	4.02	4.60	5.23	6.28	43.8	18.5	9.02	2.1	4.9
	6	3.26	3.70	4.36	5.36	44.8	12.9	5.13	3.7	8.7
	8	2.89	3.20	3.87	4.85	55.9	12.4	4.38	4.8	12.8

TRE+ incrementa el rendimiento de TRE entre dos y casi cinco veces para topologías Waxman y entre 1,6 y 3,7 veces para topologías Barabasi. Se obtiene mayor beneficio cuanto mayor es el número de nodos y mayor es el grado medio de la red. La longitud de los caminos se acorta entre un 10 y 15%, aproximándose a la longitud media de SP.

TRE+ tiene menor complejidad computacional que SP, ya que el número de comparaciones necesarias para reenviar la trama en TRE+ es d^2, mientras que en SP es $(n-1)$, siendo d el grado medio y n el número total de nodos en la red. Para una red con grado medio $d=4$, formada por 64 nodos, TRE+ solo necesita 16 comparaciones de media, mientras que SP requiere 63. Si la red se compone de 128 nodos, cada uno de ellos necesita 127 búsquedas en la tabla para reencaminar la trama, por los 16 de TRE+.

4.5 Generalización a protocolos D-TRE (*Dynamic TRE*)

La evolución conceptual de los protocolos TRE y TRE+ conduce a los protocolos D-TRE ("*Dynamic Tree-based Routing Ethernet*"). Mientras que en TRE y TRE+, el objetivo era extender el conocimiento de la red más allá del árbol. Por lo tanto, los primeros pasos del protocolo coinciden con ambos, esto es, la construcción del árbol y la asignación de direcciones locales a los nodos. La principal diferencia radica en el alcance del conocimiento de la red que posee cada nodo y el modo de conseguir dicha información.

En TRE los conmutadores utilizan el conocimiento de sus vecinos (nodos directamente conectados) que no pertenecen al árbol, es decir, a un salto de distancia. No hay intercambio de mensajes de control, únicamente descubrimiento de vecinos.

En TRE+ la información que mantienen los conmutadores de la red aumenta a dos saltos de distancia. En este protocolo se necesita un intercambio de mensajes tipo vector-distancia para descubrir los nodos situados a dos saltos (vecinos de vecinos del conmutador).

En D-TRE no se especifica el alcance del conocimiento topológico sobre la vecindad de los nodos de forma fija, como se establece en sus protocolos predecesores, sino que cada nodo calcula su alcance de forma autónoma. Algunos nodos no necesitan ampliar la información de la red, ya que ello no va a redundar en un incremento del rendimiento, ni va a modificar la decisión de reenvío que se adopta con una extensión menor. Sin embargo, el incremento del descubrimiento de la topología para otros nodos significa conseguir una mejor solución en el reenvío de las tramas gracias al uso de atajos, obteniendo rutas más cortas y optimizando el reparto de tráfico en los enlaces de la red.

Para ampliar el conocimiento parcial que cada nodo posee de la red con una extensión variable, los nodos intercambian mensajes de control de tipo "estado de enlaces".

El objetivo de un conmutador D-TRE es recopilar la información topológica necesaria, con el fin de optimizar la decisión de reenvío de una trama para conseguir mayor rendimiento en la red, pero llegando a un compromiso con el coste de adquirir, mantener y procesar dicha información. Este coste se plasma en la cantidad de información de estado a almacenar en cada conmutador y a consultar en cada reenvío. El aumento del conocimiento de la red en cada conmutador implica siempre la obtención de una decisión más optimizada (mejor o igual) en el número de saltos, que la que se toma con menos información. No obstante, el aumento progresivo del conocimiento llevaría a un descubrimiento total de la red, degenerando en un protocolo s*hortest path* y, por lo tanto, penalizando exponencialmente el coste cuanto mayor es el número de nodos de la red.

En definitiva, la extensión del conocimiento topológico que adquieren los conmutadores D-TRE se calcula dinámicamente. Además, es necesario que el cómputo de la extensión que alcanzan los conmutadores pueda calcularse utilizando parámetros conocidos por dichos conmutadores, sin que sea necesario un intercambio de tramas adicional informando a cada *conmutador* de su extensión.

Con este enfoque se han diseñado y evaluado dos protocolos D-TRE. En el primero de ellos, D-TRE1, el alcance del conocimiento topológico de un nodo dependerá de su grado (del número de vecinos que tenga), basándonos en la idea de que un nodo que tiene muchos vecinos probablemente dispone ya de suficiente información y no necesita ampliar su alcance. En el segundo, D-TRE2, el alcance de un nodo será función de su posición en el árbol de expansión que se calcule. Los nodos que se sitúen cerca de la raíz dispondrán de muchos enlaces candidatos a convertirse en atajos ya que en los niveles superiores del árbol es donde se concentra la mayoría de los enlaces y, por lo tanto, de atajos, por lo que es fundamental ampliar el alcance en estos niveles. Por el contrario, los nodos

situados lejos de la raíz apenas ganarán conocimiento por el hecho de aumentar su alcance en uno o dos saltos.

4.5.1 Aspectos comunes de los protocolos D-TRE

Ambos protocolos comparten un esquema de funcionamiento común que se puede resumir en los siguientes puntos:

1. Construcción del árbol: La red adapta la topología en un árbol de expansión mediante el protocolo STP/RSTP. La construcción del árbol se realiza igual que en los protocolos TRE y TRE+.

2. Asignación de la dirección HLMAC: Se asigna a cada nodo del árbol una dirección jerárquica local mediante el protocolo HURP [Iba+08b], al igual que en TRE y TRE+.

3. Gestión de la información topológica: Los conmutadores D-TRE intercambian mensajes de control para adquirir información de la red y construir sus tablas de reenvío, con el fin de encaminar tramas por enlaces no pertenecientes al árbol.

    - Intercambio de mensajes: Los nodos generan y reenvían mensajes de control con un formato común.

    - Construcción y actualización de las tablas de reenvío: Cada nodo construye las tablas para el reenvío de tramas de datos en función de los mensajes de control recibidos, según se extienda su conocimiento de la red.

4. Procedimiento de reenvío de tramas de datos: Elección del puerto por el que un nodo reenvía una trama de datos recibida para alcanzar un nodo destino.

5. Protocolos particulares D-TRE: El alcance del conocimiento topológico de un nodo y la gestión del intercambio de los mensajes de control será específico según el protocolo que se esté utilizando en la red, D-TRE1 o D-TRE2, y se describe en las siguientes secciones.

4.5.2 Gestión de la información de control

La familia de protocolos TRE utiliza, además del árbol, enlaces no pertenecientes a éste, es decir, hace uso de atajos para encaminar tramas. Para ello, los conmutadores deben conocer cierta información topológica que les permita posteriormente decidir si utilizar dichos atajos o reenviar la trama por la rama del árbol correspondiente al destino, ascendente o descendente.

Un conmutador está formado por un número N de puertos activos, tantos como vecinos tenga. Tras la formación del árbol, un nodo de la red dispone de un único puerto raíz, R, para encaminar las tramas hacia el "padre", y de tantos puertos designados, D, como ramas descendentes partan de dicho nodo. El resto de puertos vecinos, A, no son utilizados si el encaminamiento se efectúa estrictamente por el árbol. El objetivo de la arquitectura TRE y, por lo tanto, de D-TRE, es utilizar estos puertos cuando constituyen atajos que obtienen menor coste en los caminos y desviar así tráfico fuera del árbol.

El número de puertos vecinos (ajenos al árbol) será:

$$A = N - D - 1$$

En la Ilustración 35 se muestra una sencilla topología, a modo de ejemplo. Se observa como el conmutador '1.2' tiene seis vecinos y, por lo tanto, seis puertos *(N=6)*, de los cuales uno es el puerto raíz (*R=1* siempre) y otro es un puerto designado *(D=1)* con identificador '1', a través del cual se envían tramas por la rama descendente hasta el nodo '1.2.1', con lo que *A = 6-2 = 4* puertos vecinos. El resto corresponde con los atajos que se pueden usar en TRE para conseguir mayor rendimiento en la red siempre que proporcionen menor coste que a través del árbol. Así, por ejemplo, las tramas con destino al conmutador '2.3' deberán ser encaminadas por el puerto A2 obteniendo un coste de dos saltos [1.2 – 2 – 2.3] en lugar de cuatro, coste resultante si se encaminaran por el puerto raíz siguiendo la ruta del árbol [1.2 – 1 – 0 – 2 – 2.3]. Para alcanzar el conmutador '2.1.1', el puerto elegido deberá ser el A4, ya que el coste total es uno (es vecino), en lugar de cinco a través del árbol.

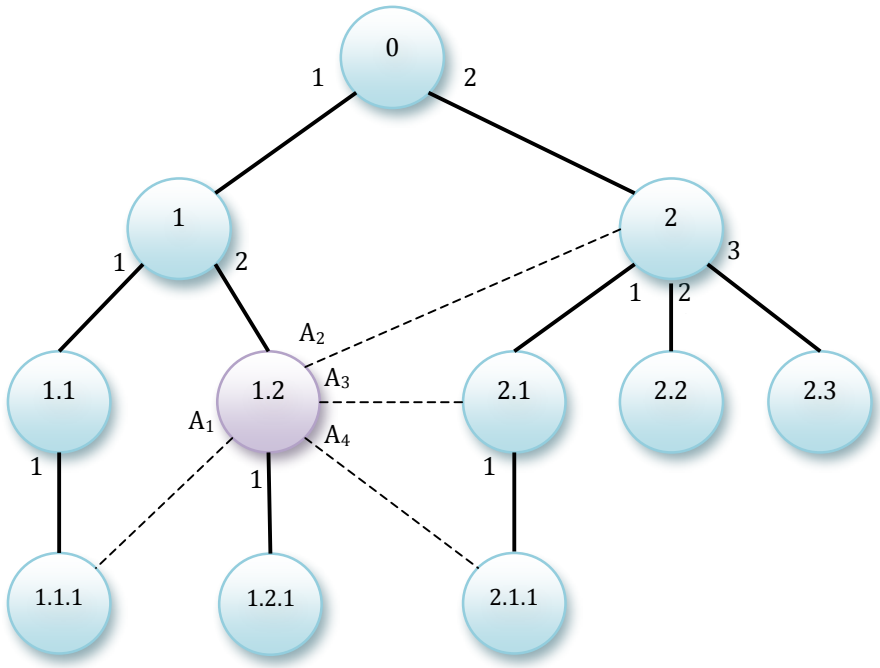


Ilustración 35. Árbol y vecindad de un conmutador

El uso de atajos en D-TRE se restringe exclusivamente a los casos en los que se obtenga a través de ellos un coste menor que siguiendo el árbol. Por consiguiente, es preciso realizar una comparación entre el coste del árbol y el proporcionado por los vecinos, siendo necesario tener conocimiento de cuáles son los vecinos que hay que incluir en el cómputo. Es decir, cada nodo necesita adquirir información de los nodos con conectividad próxima, por qué puerto alcanzarlos y a qué distancia están.

Los conmutadores D-TRE intercambian mensajes del tipo "estado de enlaces" para distribuir información de sus vecinos, con el fin de que los conmutadores receptores actualicen sus tablas de reenvío con algunos nodos alcanzables en un cierto número de saltos, considerándolos en el cálculo del coste de la ruta hacia el destino si se encaminan las tramas hacia ellos.

El objetivo es determinar qué alcance del conocimiento de vecindad es el apropiado, es decir, cuál es el límite de saltos máximo hasta donde los conmutadores deben poseer información, para posteriormente tener en cuenta a los nodos situados hasta esa distancia en la operación de decisión de reenvío, en el cálculo del coste de rutas.

En los algoritmos de tipo *shortest path* se conoce la red al completo, el conocimiento de los vecinos se extiende a la totalidad de la red, se podría decir que el alcance de un nodo es infinito (considerando el infinito como el número mayor de saltos en la red). En cambio, en los algoritmos STP/RSTP dicho alcance sería cero, ya que una vez formado el árbol se sigue estrictamente por éste sin considerar vecinos.

En TRE los conmutadores pueden utilizar los atajos que sus vecinos directos les proporcionan, es decir, tienen conocimiento con alcance uno, nodos situados a un salto de distancia. Los conmutadores TRE no necesitan intercambiar información adicional a la de la formación del árbol, ya que todos los nodos conocen cuáles son sus vecinos y qué puerto utilizar para alcanzarlos. Si un vecino pertenece a la misma rama donde está situado el destino, se hace uso del atajo proporcionado por dicho vecino, ya que el coste siempre va a ser menor que a través del árbol. Por ejemplo, en la Ilustración 35, el conmutador '1.2' al enviar una trama a '2.2' evalúa a sus vecinos: '1.1.1', '2', '2.1' y '2.1.1' y obtiene que el conmutador '2' pertenece a la misma rama que el destino (misma parte común de la dirección), por lo que encamina la trama a través del puerto A_2.

En TRE+ los conmutadores pueden utilizar la información de los nodos vecinos de vecinos (situados a dos saltos de distancia), es decir, tiene un conocimiento de alcance dos. En este caso, es necesario un mecanismo de detección de los conmutadores vecinos de los vecinos directos. Para ello se emplea el procedimiento descrito en el apartado 4.4.2, en resumen, un vector distancia limitado a un salto. Todo conmutador envía un mensaje a cada uno de sus vecinos informando de cuáles son todos sus vecinos, notificando distancia uno. Cuando un enlace se cae, se comunica distancia dos, considerada como distancia infinita en este protocolo. El conmutador que recibe la información añade, al coste de un nodo notificado en el vector-distancias, que siempre será uno, la distancia al nodo que ha comunicado dicho coste, que también será uno puesto que es vecino directo. Por ejemplo, en la Ilustración 35 el nodo '2.1' transmite el vector de distancias que aparece en la Tabla 23. Cuando el nodo '2' lo recibe, añade a cada una de las distancias el coste empleado en alcanzar al nodo emisor, siempre uno, deduciendo que, por ejemplo, para alcanzar al nodo '1.1.1' hay distancia dos a través del nodo '1.2'.

Tabla 23. Vector distancias enviado por un conmutador TRE+ (conmutador '1.2' de la Ilustración 35)

Nodo	Distancia
1.1.1	1
1.2.1	1
2	1
2.1	1
2.1.1	1

El alcance de los protocolos descritos anteriormente es fijo. Si se continúa aumentando el alcance de manera estática (alcance tres en un hipotético 3TRE, cuatro en un hipotético 4TRE, etc.), se alcanzaría la máxima distancia para algunas redes, considerada infinito, degenerando en *Shortest Path*. Es probable que no se obtuviera menor coste en la ruta, porque ya se dispone de la información óptima, y en cambio el rendimiento de la red sería peor, debido a que los conmutadores realizarían un exceso de intercambio de mensajes y de consultas a las tablas.

En D-TRE se computa dinámicamente, es decir, varía según las características del nodo particular, dependiendo del protocolo que se emplee:

- En el protocolo D-TRE1, se tiene en cuenta el número de vecinos del nodo, el grado.
- En el protocolo D-TRE2, depende del nivel del árbol en el que se sitúa el conmutador.

En la Tabla 24 se muestra una comparativa del alcance de la información según el protocolo. Un nodo dado deberá adquirir información de los nodos que estén al número de saltos que indica la columna "Alcance" y, por consiguiente, tendrá que intercambiar mensajes para distribuir su propia información de vecindad hasta esa extensión. En los dos métodos de D-TRE el alcance mínimo para todo *conmutador* es igual a uno, ya que es un protocolo de la familia TRE y por lo tanto deberá estudiar posibles atajos para decidir y realizar el encaminamiento.

Tabla 24. Comparativa de la extensión de la información según protocolo

Protocolo	Alcance
STP/RSTP	0
TRE	1
TRE+	2
3-TRE (hipotético)	3
n-TRE	n
Shortest Path	∞
…	…
D-TRE1	Variable (función del grado)
D-TRE2	Variable (función del nivel)

El protocolo de enrutamiento de estos mensajes de control en D-TRE para intercambiar información topológica será del tipo "estado de enlaces", en contraposición del empleado en TRE+, que era del tipo "vector distancias". La utilización de este mecanismo en D-TRE se justifica para prevenir la presencia de bucles y evitar la cuenta a infinito si falla algún enlace, ya que la información que un nodo posee de la red se da a conocer a más de un salto de distancia.

4.5.3 Mensajes de control

El procedimiento de descubrimiento de la topología de la red se realiza mediante el intercambio de mensajes de control entre los conmutadores que la componen. Los mensajes de control son gestionados por una subcapa del nivel de enlace superior a Ethernet. Esta subcapa desempeña una función similar a LLC, "*Logical Link Control*", (descrita en el Estándar IEEE 802.2), pero modificada y simplificada para el protocolo D-TRE.

El protocolo de enrutamiento de los mensajes de control es del tipo *estado de enlaces*, por lo que cada conmutador deberá incluir en el paquete de control del enlace de D-TRE, D-TLC ("D-TRE Link Control") cuáles son sus vecinos. Este mensaje será encapsulado (Ilustración 36) en el campo "Datos" de una trama Ethernet, cuyo campo *Tipo* (*Ether Type*) tendrá un valor de específico reservado para este protocolo.

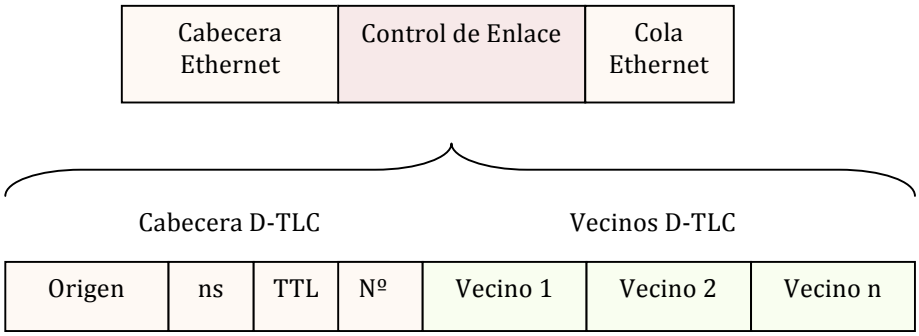


Ilustración 36. Encapsulado de la capa de control del enlace en D-TRE

La estructura del paquete D-TLC es siempre la misma, independientemente del protocolo D-TRE que se use y de las características del nodo. El formato de los mensajes se muestra en la Ilustración 36 y se detalla, campo a campo, a continuación.

Los conmutadores D-TRE reenviarán estos mensajes cada cierto tiempo. La periodicidad de envío de estos mensajes se estudia en el apartado 4.5, dedicado al análisis de la sobrecarga y la estabilidad de D-TRE.

Tabla 25. Formato de mensaje de control D-TRE

Nº octeto	Octeto impar	Octeto par
1	HLMAC origen	
3		
5		
7	Nº secuencia	
9	Nº de vecinos	TTL
11	HLMAC vecino 1	
13		
15		
17	HLMAC vecino 2	
19		
21		
10+*n*	HLMAC vecino *n*	
12+*n*		
14+*n*		
	Relleno	

Los campos del mensaje D-TLC son los siguientes:

- HLMAC ORIGEN: Indica la dirección HLMAC del conmutador que genera el mensaje.
- Nº SECUENCIA: Cada vez que un conmutador envía su "estado de enlace", incrementa en uno este campo. De esta manera, si un conmutador recibe dos mensajes de un mismo origen, se quedará con el de número de secuencia mayor, que corresponderá con el más actual.

- Nº VECINOS: Refleja el número de vecinos que tiene el conmutador emisor. Este campo es necesario para determinar el tamaño útil del paquete, ya que el mensaje tiene un tamaño variable (Ilustración 36).

- TTL: Define el alcance del mensaje. Cada conmutador que recibe el mensaje disminuye su valor antes de reenviarlo. Cuando el TTL de un mensaje llega a 0, el conmutador lo descarta y no lo reenvía. La gestión de este campo por parte del conmutador depende del protocolo D-TRE utilizado, y se concretará para cada protocolo en las siguientes secciones. En todo caso:

    - Si un conmutador decrementa el valor del campo TTL hasta cero, el mensaje se descarta sin más. Nunca se reenviará un mensaje con TTL=0.

    - El valor máximo del TTL es dos. En este caso, un conmutador difunde sus vecinos directamente conectados a conmutadores situados a dos saltos de distancia. Por lo tanto, el conmutador que recibe este "estado de enlaces", obtendrá información de los vecinos de un conmutador situado a dos saltos de distancia, es decir, descubrirá conmutadores situados a tres saltos de distancia, extensión máxima estipulada para los dos métodos del protocolo D-TRE.

- VECINOS: Cada vecino tiene un tamaño de 6 octetos, por lo que el tamaño total del mensaje será: nº vecinos x 6 + 10 (+ relleno, si es menor de 46 octetos)

- RELLENO: Al ir encapsulado en el campo "Datos" de una trama Ethernet, debe ser de al menos 46 octetos, mínimo tamaño de dicho campo y, además el tamaño múltiplo de 8 octetos.

4.5.4 Actualización de la información topológica

La información de control recibida en forma de mensajes D-TLC se gestiona de forma general para todos los conmutadores D-TRE como se indica a continuación:

Los conmutadores D-TRE disponen de tres herramientas para adquirir y gestionar la información de control recibida y conseguir así encaminar una trama de datos de una manera óptima.

- Tabla de Enlaces: Tabla existente en cada conmutador donde se guardan los mensajes de control recibidos de cada nodo origen. En ella se almacena la información existente en la última versión de los mensajes D-TLC recibidos de cada nodo. Se usa para detectar cambios en los enlaces y para configurar la Tabla de Reenvío.

- Tabla de Reenvío: En el momento de realizar el encaminamiento de una trama, todo conmutador D-TRE consultará su Tabla de Reenvío para determinar el puerto por el que debe ser enviada en función de la dirección del destino.

- Algoritmos de Actualización Topológica: Algoritmo que se ejecuta cada vez que se recibe un mensaje de control D-TLC para actualizar la Tabla de Enlaces y, periódicamente, para actualizar la Tabla de Reenvío.

Cada conmutador recibe periódicamente mensajes de control con la estructura de la Tabla 25 y los almacena en su Tabla de Enlaces. Su utilidad es detectar cuándo se producen cambios o fallos en

los enlaces de la red que puedan afectar a la decisión de reenvío correspondiente. La tabla está compuesta de cuatro campos:

- Nodo origen: Se corresponde con el nodo origen del mensaje de control recibido, el nodo que ha generado el mensaje D-TLC dado.
- Puerto recibido: Indica el puerto por el que se ha recibido el mensaje. El reenvío de un mensaje recibido no se realiza a través del puerto por el que se ha recibido el mensaje para evitar redundancia.
- Nº secuencia: Se corresponde con el campo "número de secuencia" del mensaje de control.
- Vecinos: Se almacenan las direcciones HLMAC de todos los vecinos que se incluyen en el mensaje.

Cuando se recibe un mensaje de control D-TLC, se añade una entrada en la Tabla de Enlaces y se activa un temporizador para controlar el tiempo de validez de dicha entrada. La tabla se actualiza como sigue: si se recibe un mensaje de un nodo origen por el mismo puerto que el asociado en la tabla para dicho nodo, se modificará la entrada en la tabla si algún vecino difiere y el número de secuencia del mensaje es mayor (más actual).

Tabla 26. Tabla de Enlaces de un conmutador D-TRE

Nodo origen	Puerto recibido	Nº Secuencia	Vecinos		
N	P_{RX}	ns	V_1	V_2	V_n

4.5.4.1 Tabla de Reenvío

La Tabla de Reenvío se consulta cada vez que se recibe una trama de datos, para conocer el puerto por el que se ha de encaminar, obteniendo la ruta más corta para el destino que figure en dicha trama de datos. La tabla está compuesta de tres campos:

- Nodo: Identifica a un conmutador particular de la red. En este campo se realizará la comparativa con la dirección destino existente en la trama.
- Siguiente salto: Indica el puerto por el que se ha de reenviar la trama para alcanzar el "nodo".
- Distancia: Indica el número de saltos empleados en alcanzar el "nodo" transmitiendo la trama a través del "siguiente salto".

Tabla 27. Tabla de Reenvío de un conmutador D-TRE

Nodo	Siguiente salto	Distancia
N	S	d

Todo conmutador construye inicialmente su Tabla de Reenvío con sus vecinos y la actualiza con la información extraída de los paquetes de estado que recibe, almacenada en la Tabla de Enlaces. La Tabla de Reenvío no tiene caducidad, se reinicia cuando se detecta un fallo o un cambio topológico del árbol y su periodo de actualización coincide con la periodicidad de envío (y por lo tanto de recepción) de mensajes de control D-TLC.

4.5.4.2 Algoritmo de actualización topológica

Siempre que un conmutador recibe un mensaje de control D-TLC realiza dos tareas: actualizar la información topológica que posee y reenviar o descartar el mensaje. El reenvío del mensaje se realizará dependiendo del método utilizado y siempre que el campo TTL sea mayor que cero.

La actualización topológica se lleva a cabo con el siguiente algoritmo se describe a continuación y se puede observar en el diagrama de flujo de la Ilustración 37.

En primer lugar, realiza una búsqueda en la Tabla de Enlaces, el par de campos <Nodo Origen, Puerto Recibido> para averiguar si ya existe en la tabla:

- Si no existe, crea una entrada para dicho nodo, con la información presente en el mensaje de control (número de secuencia y vecinos) y especifica el puerto por el que ha recibido el mensaje. Debido a que se ha realizado una actualización de la información topológica en la Tabla de Enlaces, se invoca al algoritmo de actualización de la Tabla de Reenvío.

- Si ya existe el par <Nodo Origen, Puerto Recibido> en la Tabla de Enlaces, comprueba si el número de secuencia de la entrada en la tabla para ese nodo es mayor o menor que el recibido en el mensaje.

    - Si es mayor, comprueba si hay alguna diferencia entre los vecinos incluidos en el mensaje y los existentes en la tabla.

        - Si hay diferencia, modifica los vecinos de la tabla, estableciendo los que aparecen en el mensaje. En este caso se procederá a actualizar la Tabla de Reenvío, porque ha habido modificación de algún vecino del nodo origen.

        - Si no hay diferencia en los vecinos, únicamente actualiza el número de secuencia de la tabla.

    - En caso contrario, se descarta el mensaje.

Si una entrada de la Tabla de Enlaces caduca, se entiende que su mensaje correspondiente no se va a volver a recibir y, por lo tanto, se han producido cambios en la topología de la red. Se invoca al algoritmo de la Tabla de Reenvío y se pone en conocimiento de un plano superior para interpretar el cambio.

Hay que tener en cuenta que la existencia de diferencias en los vecinos y la no recepción de los mensajes pueden ser motivadas por un fallo, por lo que se deberá analizar esta situación. También hay que conocer la periodicidad de envío/recepción de los mensajes y cuánto tiempo tiene que pasar para que se confirme su no recepción y la caducidad de la tabla. Ambos factores se analizan en la sección de *Estabilidad*.

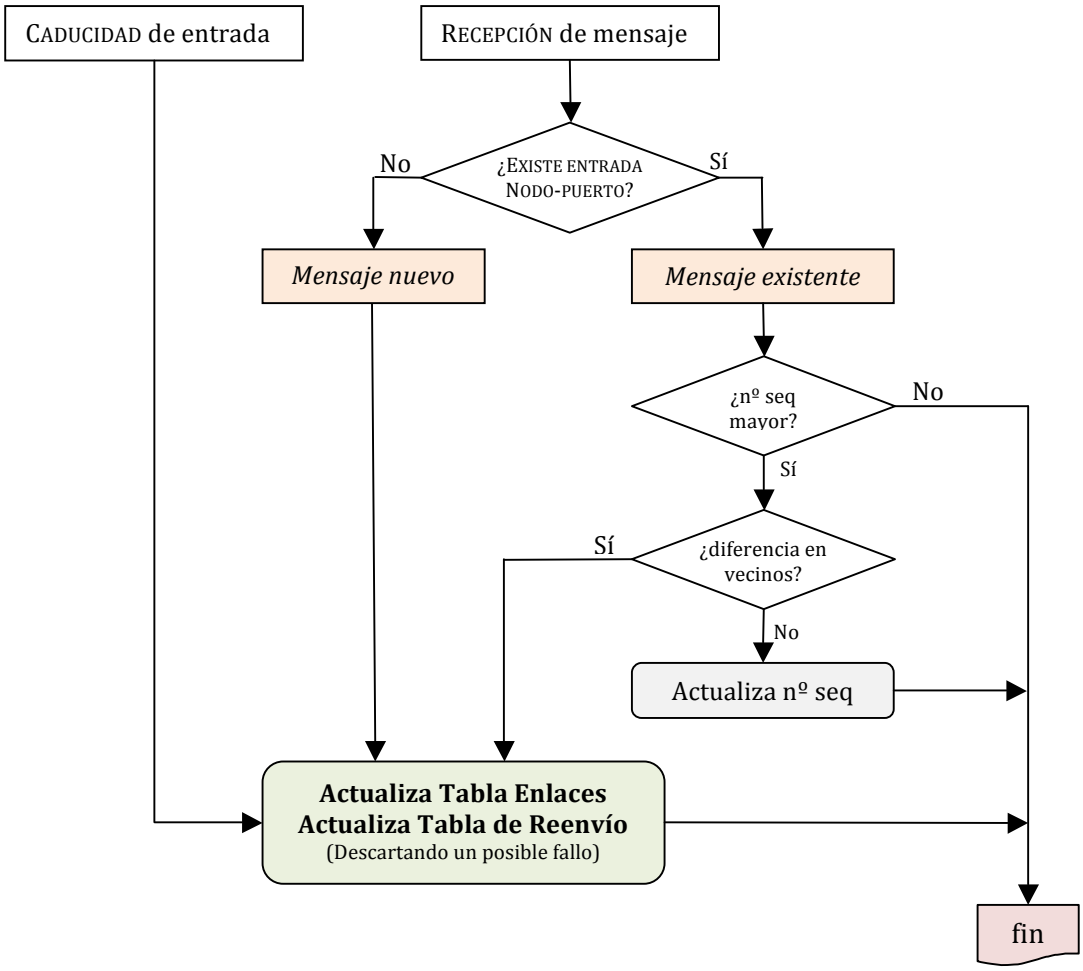


Ilustración 37. Algoritmo de actualización topológica (Tabla de Enlaces)

4.5.4.3 Actualización de la Tabla de Reenvío

Este procedimiento consta de tres pasos:

1. Incluir a los nodos situados a un salto (mis vecinos): En primer lugar, se inicia la tabla con los vecinos del *conmutador*. Los primeros registros de la Tabla de Reenvío son los formados por los vecinos, situados a un salto de distancia y conectados con el conmutador por su enlace directo, por lo que el siguiente salto son ellos mismos, con distancia uno.

2. Incluir a los nodos situados a dos saltos (vecinos de mis vecinos): En segundo lugar, se analizan cada una de las entradas de la Tabla de Enlaces que corresponden a los vecinos directos, es decir, se analizan los mensajes difundidos por los vecinos directos, los cuales se acaban de añadir a la tabla con distancia uno y que poseen vecinos situados a dos saltos del nodo en estudio. Se ignoran los conmutadores ya existentes en la tabla (los que son vecinos a un salto y el propio conmutador). El resto de vecinos se añade a la Tabla de Reenvío con distancia igual a dos, y el vecino origen del mensaje como siguiente salto.

3. Incluir nodos situados a tres saltos (vecinos de los vecinos de mis vecinos): En tercer lugar, y por último, se analizan cada una de las entradas de la Tabla de Enlaces que corresponden a los vecinos de los vecinos, es decir, se analizan los mensajes difundidos por los nodos situados a dos saltos, que se acaban de añadir a la tabla con distancia igual a dos, y que poseen vecinos situados a tres saltos del nodo en estudio. Se ignoran los conmutadores ya existentes en la tabla (los que son vecinos a un salto y dos saltos y el propio conmutador). El resto de vecinos se añade a la Tabla de Reenvío con distancia tres. El siguiente salto no puede ser el nodo origen que se está estudiando, porque no tiene conectividad con el conmutador (están a distancia dos), por lo que será el mismo "siguiente salto" que para alcanzar a éste, ya incorporado en la Tabla de Reenvío.

Siempre que se produzcan modificaciones en los enlaces (descubiertas mediante recepción de mensajes nuevos, diferencia de vecinos en los mensajes ya recibidos o caducidad de las entradas de la Tabla de Enlaces, como se observa en la Ilustración 37), se actualizará la Tabla de Enlaces y se ejecutará este algoritmo para actualizar la Tabla de Reenvío.

4.5.5 Reenvío de tramas de datos

Una vez creada la Tabla de Reenvío del conmutador D-TRE, ya está preparado para encaminar tramas de datos. El procedimiento de reenvío de datos consiste en elegir el puerto idóneo (siguiente salto) para encaminar una trama en función de la dirección destino que figure en la misma, consiguiendo el menor coste posible. Este procedimiento, que es el mismo para los dos protocolos D-TRE, se realiza en los pasos que se muestran en el diagrama de flujo de la Ilustración 38 y se describen a continuación:

Para toda trama que haya que enviar, se realiza una comparación de la HLMAC del conmutador origen, que reenvía la trama, con la HLMAC del conmutador destino para determinar si están situados en la misma rama o en ramas diferentes.

1. Si la dirección de uno de los conmutadores está contenida en la del otro, es decir, comparten prefijo (ambos conmutadores comparten bytes de sus respectivas direcciones, comenzando por los bytes de mayor peso y hasta que aparezca un byte igual a cero en alguna de ellas), implica que están situados en la misma rama del árbol. En este caso se sigue estrictamente el camino del árbol, que siempre será el óptimo. La distancia será la diferencia del número de bytes distintos de cero presentes en ambas direcciones, o calculándolo de otra manera, la diferencia del número de ceros en cada una de las direcciones (siempre en valor absoluto).

    a. Si el primer "cero" aparece en la HLMAC del nodo origen, indica que están situados en la misma rama y que el nodo destino está situado en alguna de sus ramas descendentes. El puerto elegido para el encaminamiento coincide con el siguiente dígito distinto de cero de la dirección destino.

       Por ejemplo, si el nodo origen '2.1.0.0.0.0' tiene que enviar una trama al nodo '2.1.1.2.0.0', comprueba si ambos pertenecen a la misma rama, es decir, si tienen el mismo prefijo (coinciden exactamente sus primeros dígitos) hasta que aparece el primer cero en una de ellas. En este caso se cumple, ya que los dos primeros dígitos, "2.1", coinciden en ambos. Como el primer cero aparece en la dirección origen, el destino está en la rama descendente. El puerto elegido será el puerto designado cuyo número sea igual al siguiente dígito de la dirección destino distinto de cero, en este caso "1" hasta llegar al nodo '2.1.1.0.0.0', que hará la misma operación y encaminará la trama hacia '2.1.1.2.0.0'. La distancia entre ambos conmutadores es el número de elementos

distintos de cero, una vez eliminado el prefijo común, por lo que están situados a dos saltos de distancia.

b. Si el primer "cero" aparece en la HLMAC del nodo destino, indica que están situados en la misma rama y que el nodo destino está situado en su rama superior. El puerto elegido para el reenvío será el puerto raíz.

   Por ejemplo, si el nodo origen' 2.1.1.2.0.0' tiene que encaminar una trama al nodo '2.1.0.0.0.0', y una vez confirmado que en ambos coinciden exactamente sus primeros dígitos hasta el primer cero, se comprueba que este primer cero aparece en la dirección del nodo destino, por lo tanto está situado en su rama superior. El puerto elegido será el puerto raíz. El nodo origen '2.1.1.2.0.0' encamina la trama por su puerto raíz hasta '2.1.1.0.0.0' y, éste a su vez, realiza la misma operación hasta llegar a '2.1.0.0.0.0'. Igual que en el párrafo anterior se calcula la distancia, siendo de dos saltos.

2. Si las direcciones de los nodos origen y destino no comparten prefijo (difieren al compararlas desde el octeto de mayor peso, de izquierda a derecha en la notación por puntos, encontrándose dos dígitos diferentes entre sí, y siendo ambos distintos de cero), implica que están situados en diferentes ramas del árbol. En este caso, el algoritmo trata de encontrar un atajo que proporcione menor coste que el camino seguido por defecto a través del árbol.

   a. Primero se calcula la distancia al destino siguiendo el árbol: En primer lugar se elimina del cómputo la parte igual de ambas direcciones si se diera el caso, comparando como siempre de izquierda a derecha. En segundo lugar se ignoran los ceros de ambas direcciones, que aparecerán siempre en las últimas posiciones de cada dirección. Por último, se suman los dígitos restantes de ambas direcciones, obteniendo la distancia entre ambos nodos siguiendo el árbol.

   b. Después se averigua si existe algún atajo que proporcione una ruta más corta que la obtenida por defecto siguiendo el árbol. Para ello se consulta en la Tabla de Reenvío si existe alguna entrada que proporcione menor coste que la calculada por defecto en el apartado anterior. Se buscan entradas que pertenezcan a la misma rama del nodo destino, o el propio destino, y que la distancia sea menor que la del árbol.

      i. Si el nodo destino aparece en la Tabla de Reenvío, la distancia será exactamente la que aparece en el campo "distancia" de dicha tabla.

      ii. Si no aparece el nodo destino, pero sí un nodo de su misma rama, la distancia a través del atajo se obtiene sumando el campo "distancia" de la tabla hasta ese nodo, más la distancia desde éste al nodo destino, calculado como se indica en el punto 1, restando el número de bytes distintos de cero presentes en ambas direcciones.

   c. Después de realizar las comparaciones anteriores, para encaminar la trama se escoge de la Tabla de Reenvío la entrada que proporciona la menor distancia (siempre que sea menor que la ruta por defecto a través del árbol). El siguiente salto será el indicado en el campo "siguiente salto" en el registro de la Tabla de Reenvío que ha proporcionado el menor coste.

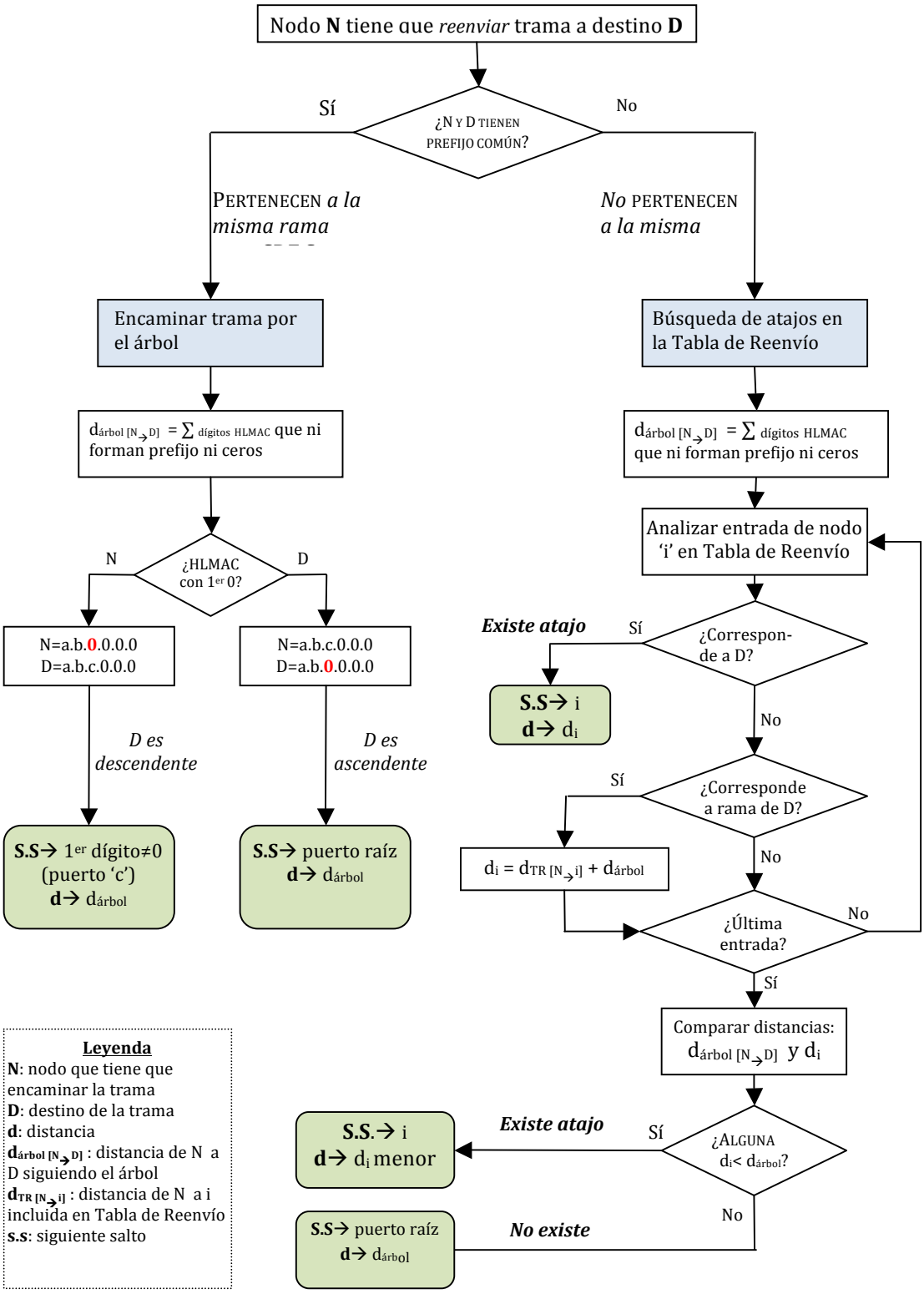


Ilustración 38. Procedimiento de Reenvío de tramas de datos en D-TRE

A continuación se muestra el procedimiento de reenvío de datos para el nodo '2.1.1' de la Ilustración 39, con la Tabla de Reenvío que se muestra en la Tabla 28, cuando tiene que encaminar una trama al nodo '2.6.8.5'.

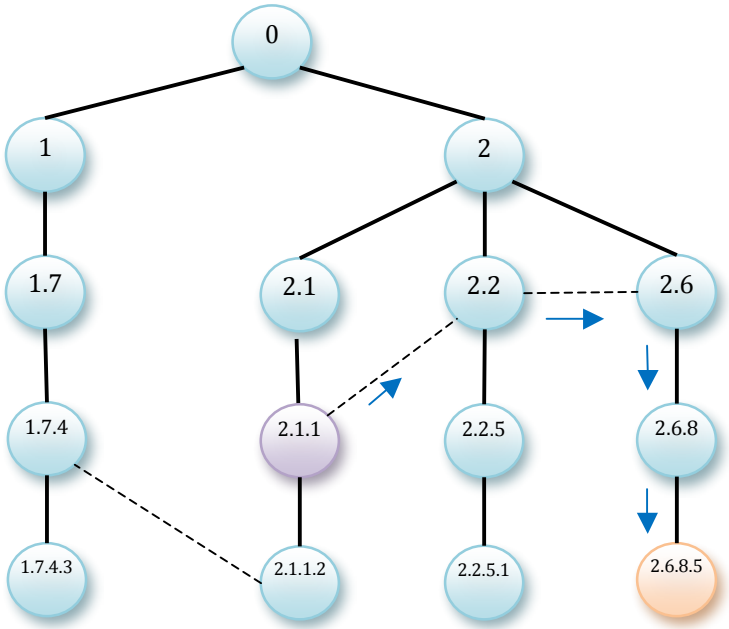


Ilustración 39. Ejemplo del Procedimiento de Reenvío de datos en D-TRE. Ruta hasta '2.6.8.5' siguiendo el atajo ofrecido por '2.2'

Tabla 28. Ejemplo de Tabla de Reenvío de nodo D-TRE '2.1.1' de la Ilustración 39

Nodo	Siguiente salto	Distancia
2.1	2.1	1
2.1.1.2	2.1.1.2	1
2.2	2.2	1
2	2.1	2
2.2.5	2.2	2
2.6	2.2	2

Primero se compara la dirección origen, '2.1.1.0.0.0' con la dirección destino, '2.6.8.5.0.0' para determinar si están en la misma rama del árbol. El primer dígito, "2", coincide, pero no los segundos dígitos y ninguno de ellos es nulo ("1" y "6"). Por lo tanto, no están situados en la misma rama.

Dado que origen y destino se encuentran en ramas diferentes del árbol, se va a intentar hacer uso de algún atajo, si existe. Para ello se hace uso de la Tabla de Reenvío para intentar localizar al menos un registro de algún nodo que proporcione menor coste que el camino seguido por defecto a través del árbol.

1. Se calcula la distancia al destino siguiendo el árbol: se elimina del cómputo la parte igual de ambas direcciones de izquierda a derecha, en este caso el primer elemento de cada dirección, el '2'. Después se ignoran los ceros de ambas direcciones, en este caso tres ceros del origen y dos del destino. Por último, se suman los dígitos restantes de ambas direcciones, '1.1' en el origen y '6.8.5' en el destino. Es decir, la ruta por defecto siguiendo el árbol tiene un coste de cinco saltos.

2. Se busca si existe alguna entrada en la Tabla de Reenvío que pertenezca a la misma rama que el destino. En este caso se encuentra un registro para '2.6' ('2.6.0.0.0.0'), que no es el nodo destino, pero está en la misma rama que él, '2.6.8.5.0.0'. Y se calcula el coste de la ruta siguiendo esta opción. Para ello se suman el campo "Distancia" de la Tabla de Reenvío para esta entrada, dos en este caso, y la distancia hasta el nodo destino siguiendo el árbol, dos (de '2.6.0.0.0.0' a '2.6.8.5.0.0' hay dos saltos). La ruta desde el nodo origen, '2.1.1', hasta el destino, '2.6.8.5', a través del atajo vía el nodo '2.6' será en total de 4 saltos, obteniendo una mejora de un salto respecto a la que se habría obtenido siguiendo el árbol.

Tras realizar todas las comparaciones posibles en la Tabla de Reenvío, se llega a la conclusión de que la ruta óptima se obtiene encaminando la trama hacia el nodo '2.6', por lo que habrá que reenviarla por el enlace que une al nodo '2.1.1' con el nodo '2.2' (el siguiente salto será '2.2').

En el anterior ejemplo se encuentra solamente un único atajo en la Tabla de Reenvío, pero podría haber varias opciones, siendo necesario comparar y elegir la que proporcione una ruta más corta. Además, no siempre encontrar una entrada en la Tabla de Reenvío implica una mejora respecto al árbol, por ese motivo se debe comparar también con la ruta por defecto.

4.5.6 Información topológica

Una vez descritos los procedimientos generales de operación de un nodo D-TRE, queda establecer a qué número de saltos es capaz de descubrir vecinos. Es decir, el objetivo es determinar el alcance del conocimiento que un nodo posee de la red para descubrir atajos y el modo en el que adquiere dicha información.

El principio de D-TRE es que los nodos de la red no poseen el mismo alcance fijo, sino que este es variable con el objetivo de optimizar el encaminamiento. Evidentemente, si se aumenta progresivamente el alcance de todos los nodos, se obtendrían cada vez rutas más cortas, degenerando al final en un protocolo de tipo *shortest path*. Partiendo de este principio básico se han diseñado y evaluado dos variantes, D-TRE-1 y D-TRE2.

En la variante que denominamos, D-TRE1, se ha considerado que un nodo con un número alto de vecinos dispondrá de información suficiente de la red para realizar un encaminamiento eficiente. Por ejemplo, si una red dispone de 32 nodos y un nodo A tiene 8 vecinos, es posible que conociendo a sus vecinos, tenga información de bastantes atajos. Sin embargo, en esta misma red, si otro nodo B solo tiene 1 vecino, necesitaría conocer nodos a más saltos de distancia para que el encaminamiento sea más efectivo. Sin embargo, el mismo nodo A, con 8 vecinos, en una red de 512 nodos puede no disponer de información suficiente para realizar un encaminamiento óptimo.

La cantidad de vecinos que debe poseer un nodo para que no necesite ampliar su alcance de conocimiento se puede fijar de forma estática para redes de cualquier tipo y tamaño, pero no sería eficiente. Por lo tanto, se ha decidido que el umbral se fije específicamente para cada red, siendo éste el grado medio de los nodos que la componen. En el protocolo D-TRE1, los nodos con grado mayor que la media de la red solo poseerán la información de la red que forman sus vecinos, mientras que el resto de los nodos descubrirán la red por cada una de sus ramas (vecinos) hasta que se encuentren con un nodo con grado mayor que la media.

Se ha diseñado de esta manera porque lo nodos con un número elevado de vecinos actúan como *hubs* de la red y disponen de una cantidad de información suficiente, situada a un salto, como para no necesitar ampliar este conocimiento en número de saltos. Sin embargo, un nodo con un número escaso de vecinos, posee poca información topológica proporcionada por dichos vecinos y, por lo

tanto, necesita ampliar su conocimiento de la red a nodos situados a mayor distancia. Este alcance se incrementará progresivamente hasta conocer a un nodo de grado mayor que la media de la red y recibir de éste sus vecinos. Así el conocimiento de la red de un nodo de grado menor que la media se ampliará, añadiendo la información topológica del nodo de grado mayor que la media más cercano por cada uno de sus enlaces.

En el segundo protocolo, D-TRE2, se intentan utilizar los atajos existentes en los niveles superiores del árbol, con el fin de liberar de carga a los enlaces cercanos a la raíz. La idea es que una trama enviada desde el nivel inferior (nivel 6), suba por la rama sin más cómputos (a no ser que el destino se consiga fácilmente) y a medida que va ascendiendo por el árbol, los cálculos se incrementan para detectar atajos que proporcionen una ruta mejor que el propio árbol y libere de carga los enlaces de la raíz.

En el protocolo D-TRE2, el alcance de conocimiento de los nodos se va incrementando a medida que se asciende por el árbol, siendo este igual a tres para los nodos de nivel 1º y 2º (los superiores), alcance igual a dos para los niveles 3º y 4º (los intermedios), y alcance igual a uno para los niveles 5º y 6º (los inferiores). En este protocolo no se distingue por número de nodos de la red, ya que el árbol tiene siempre como máximo 6 niveles, y todos los nodos de que disponga la red estarán distribuidos en esos 6 niveles.

Se ha diseñado de esta manera porque el árbol de expansión tiende a situar los nodos con grado alto (muy conectados) en los primeros niveles del árbol. Por lo tanto, los nodos situados en los primeros niveles son los mejores candidatos potenciales para proporcionar atajos en la red.

4.5.7 El protocolo D-TRE1

En el protocolo D-TRE1, el alcance de la información que adquieren los conmutadores depende del número de vecinos a los que esté conectado. En este protocolo se considera que un conmutador con un grado de vecindad alto obtiene de sus vecinos y, por lo tanto, aporta en los mensajes de control que envía, información suficiente de la red, no necesitando ampliar el conocimiento topológico.

Si el grado del conmutador es mayor que el grado medio de la red, solo se tienen en cuenta los vecinos directamente conectados (a distancia un salto), asemejando su funcionamiento al del protocolo básico TRE.

Si el grado del conmutador es menor (o igual) que el grado medio, la información que deberá manejar será extendida por sus vecinos hasta que, a través de cada uno de ellos, se descubra a un nodo de grado mayor que la media de la red. Es decir, un nodo conocerá sucesivamente a los vecinos de sus vecinos hasta que alguno de ellos tenga un grado mayor que la media de la topología, el cual aporta la información de sus vecinos, pero no seguirá expandiendo el conocimiento de la red más allá de éstos.

Se establece como alcance máximo tres saltos, la mitad del número máximo de niveles del árbol. Esto implica que los nodos de los niveles 3º y 4º pueden conocer, como máximo, todos los posibles atajos que les proporcione su rama completa, ya que, por ejemplo, desde el nivel 3º hasta el 6º (extremo inferior) existen esos tres saltos de distancia, y hasta el nivel 1º (extremo superior) hay solo dos saltos. Los nodos de los niveles 1º y 2º podrían conocer atajos hasta los niveles 4º y 5º de su rama, respectivamente, aunque normalmente la mayor concentración de vecinos se sitúa en los primeros niveles, en torno a la raíz, por lo que la mayoría de las veces descubrirán algún nodo con grado mayor que la media y no adquirirán información más allá de dicho nodo.

En la Ilustración 40 se observa una topología de grado medio *d=2,8*. Como ejemplo de operación vamos a analizar la información que debe adquirir el nodo '2.3.1.1' (resaltado en color morado). Siempre va a conocer la existencia de sus vecinos (a distancia un salto) para intentar hacer uso de los posibles atajos, en este caso '2.1.1' y '2.3.1' (nodo padre). El grado de '2.3.1.1' es dos, menor que el objetivo, por lo tanto, obtendrá información de los vecinos de su vecinos (a distancia dos saltos), como mínimo.

De sus vecinos obtiene la siguiente información:

- El nodo '2.3.1' informará de cuáles son sus vecinos, pero no seguirá más lejos, ya que tiene un grado igual a tres, mayor que la media. Transmitirá la vecindad de '1.1.1.3' y '2.3' (obviamente también transmitirá la dirección '2.3.1.1', aunque sea ignorada por el nodo origen '2.3.1.1' por ser su propia dirección). El alcance de la información obtenida a través de este vecino ha llegado hasta dos saltos. Se señalan en naranja los nodos "conocidos" a través del nodo '2.3.1.1'.

- El nodo '2.1.1' informará de cuáles son sus vecinos (en este caso solo el nodo '2.1') y además extenderá la información de los vecinos de sus vecinos, ya que tiene un grado igual a dos (menor que la media).

- A su vez, el nodo '2.1' informará de cuáles son sus vecinos, pero no seguirá más lejos, ya que tiene un grado igual a tres (mayor que la media). Indicará la vecindad de los nodos '1.2', '2' y '2.1.1'. El alcance de la información por el vecino '2.1.1' llega, por tanto, hasta tres saltos de distancia, ya que el nodo origen '2.3.1.1' ha recibido la información de '1.2' y '2', situados a tres saltos a través de '2.1.1' (aunque el nodo '2' ya era conocido por ser de su rama). Se señalan en rojo los nodos "conocidos" a través del nodo '2.1'.

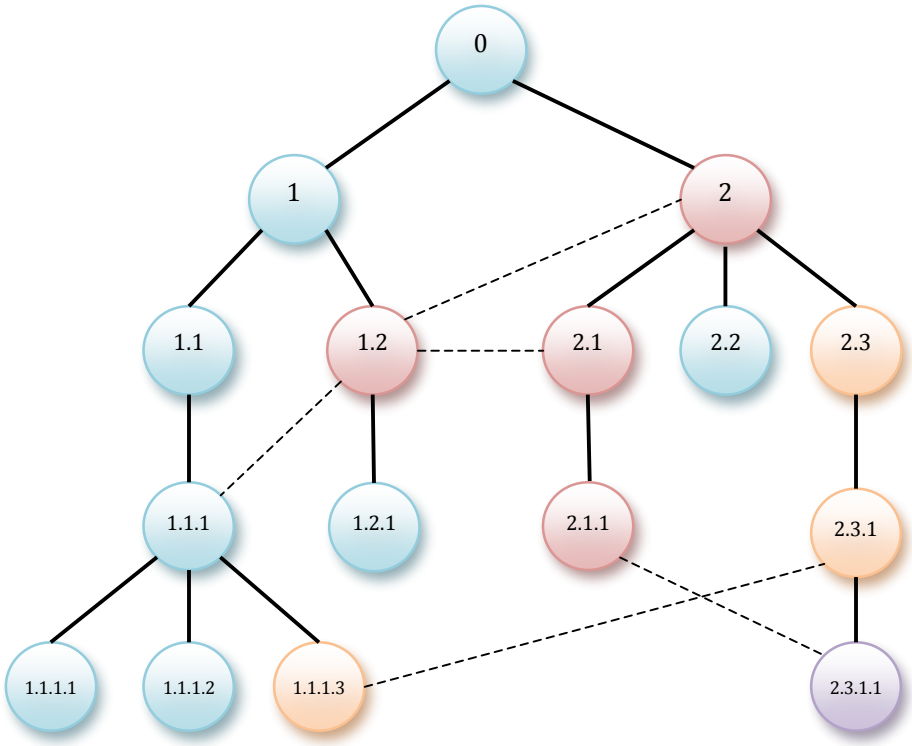


Ilustración 40. Alcance de la información topológica de un nodo D-TRE1

4.5.7.1 Intercambio de mensajes

El procedimiento de intercambio de mensajes topológicos se realiza mediante el envío de un mensaje de control D-TLC con la estructura descrita en el apartado 4.5.3

El tratamiento (preparación y reenvío) de los mensajes dependerá del valor del campo TTL con el que se controla el alcance de la información de vecindad codificada y dependerá de la relación entre su grado y el grado medio de la red:

En los nodos con grado mayor que la media:

- Si el nodo genera un mensaje: establece el valor (TTL=2) y lo envía a sus vecinos.
- Si el nodo recibe un mensaje: lo descarta, no lo reenvía, aunque el valor de TTL sea mayor que uno. Solo se generan mensajes, no se reenvían, y no actualiza su tabla.

Por ejemplo, el mensaje de estado de enlace generado por el nodo '2.1' (con un número de vecinos mayor que la media) de la Ilustración 40 será el siguiente:

<table>
<tr><td colspan="3">HLMAC</td><td colspan="3">2.1.0.0.0.0</td></tr>
<tr><td colspan="3">Nº SEQ</td><td colspan="3">Hx01</td></tr>
<tr><td>Nº VEC</td><td>Tipo</td><td>TTL</td><td>3</td><td>1</td><td>2</td></tr>
<tr><td colspan="3">VECINO1</td><td colspan="3">2.0.0.0.0.0</td></tr>
<tr><td colspan="3">VECINO2</td><td colspan="3">2.1.1.0.0.0</td></tr>
<tr><td colspan="3">VECINO3</td><td colspan="3">1.2.0.0.0.0</td></tr>
</table>

Ilustración 41. Mensaje D-TLC "estado de enlaces" enviado por un conmutador de grado mayor que la media: conmutador 2.1 de la Ilustración 40

La Tabla de reenvío de este tipo de nodos es estática (no se actualiza mediante el proceso de aprendizaje) y coincide con la de un nodo del protocolo básico TRE. En la Tabla 29 se muestra la Tabla de Reenvío para el nodo '2.1' de la Ilustración 40.

Tabla 29. Tabla de Reenvío inicial de un conmutador de grado mayor que la media: conmutador '2.1' para la red de la Ilustración 40

Nodo	Siguiente salto	Distancia
2	2	1
1.2	1.2	1
2.1.1	2.1.1	1

En los nodos de grado menor o igual que la media:

- Si el nodo genera un mensaje: establece el valor (TTL=2) y lo envía a sus vecinos.
- Si el nodo recibe un mensaje: decrementa el valor del campo TTL en una unidad y lo reenvía si sigue siendo mayor que cero, en caso contrario lo descarta. Y actualiza su tabla.

Por ejemplo, el mensaje de estado de enlace generado por el nodo '2.1.1' (con un número de vecinos menor que la media de la red) de la Ilustración 40, será el que aparece en la Ilustración 42.

HLMAC			2.1.1.0.0.0		
Nº SEQ			Hx01		
Nº VEC	Tipo	TTL	2	1	2
VECINO1			2.1.0.0.0.0		
VECINO2			2.3.1.1.0.0		

Ilustración 42. Mensaje D-TLC "estado de enlaces" enviado por un conmutador de grado menor que la media: conmutador '2.1' de la Ilustración 40

En este tipo de nodos la Tabla de Reenvío definitiva se crea/actualiza según lo descrito en el "Algoritmo de Actualización Topológica" del apartado 4.5.4.3. Cada vez que un nodo de grado menor que la media recibe un mensaje de control D-TLC de alguno de sus vecinos, se actualizan las entradas en la Tabla de Enlaces, con la que se configurará la Tabla de Reenvío. Por lo tanto, la información topológica definitiva se alcanza tras la recepción de varios mensajes.

4.5.7.2 Ejemplo de operación

En el siguiente ejemplo se detalla el proceso de gestión de la información topológica para el nodo '2.3.1.1' de la Ilustración 40. En primer lugar se analizan los mensajes recibidos. Después se crea la Tabla de Enlaces a partir de la información que contienen dichos mensajes. Y, por último, se detalla cómo se genera la Tabla de Reenvío.

Recibe y procesa los mensajes generados por sus vecinos '2.3.1.1' y '2.1.1':

El primero le comunica que tiene a los nodos '1.1.1.3', '2.3' y '2.3.1.1' a un salto (ver Ilustración 43), pero esta información es descartada al tratarse de un nodo de grado mayor que la media.

Nodo origen	**TTL**	**Vecinos**		
2.3.1	TTL=2	2.3	2.3.1.1	1.1.1.3

Ilustración 43. Mensaje generado por el nodo '2.3.1' de la Ilustración 40

Por su parte, '2.1.1' es un nodo con dos vecinos, es decir, tiene un grado menor que la media. Por lo tanto, transmite su mensaje de control con TTL=2 para informar que sus vecinos son los nodos '2.1' y '2.3.1.1'.

Nodo origen	**TTL**	**Vecinos**	
2.1.1	TTL=2	2.1	2.3.1.1

Ilustración 44. Mensaje generado por el nodo '2.1.1' de la Ilustración 40

También recibe y procesa el mensaje de control del nodo '2.1', reenviado con (TTL=1) por el nodo '2.1.1' (ver Ilustración 45)

Nodo origen	**TTL**	**Vecinos**		
2.1	TTL=1	2	2.1.1	1.2

Ilustración 45. Mensaje generado por el nodo '2.1' y reenviado por '2.1.1' (Ilustración 40)

Con cada mensaje recibido se crea un registro por cada nodo origen con la información recibida. Si se recibiera un mensaje con un número de secuencia menor que el presente en la tabla, se descartaría y no se actualizaría la Tabla de Reenvío. Así se completa la Tabla de Enlaces que se muestra en la Tabla 30.

Tabla 30. Tabla de Enlaces

Nodo origen	Puerto recibido	Nº Secuencia	Vecinos		
2.3.1	2.3.1	1	2.3	2.3.1.1	1.1.1.3
2.1.1	2.1.1	1	2.1	2.3.1.1	
2.1	2.1.1	1	2	2.1.1	1.2

A partir de esa información se genera la Tabla de Reenvío. Primero se incluye únicamente a los vecinos dando lugar a la Tabla de Reenvío *inicial* mostrada en la Tabla 31.

Tabla 31. Tabla de Reenvío inicial del nodo '2.3.1.1' de la red de la Ilustración 40)

Nodo	Siguiente salto	Distancia
2.3.1	2.3.1	1
2.1.1	2.1.1	1

Después se incluye a los nodos situados a dos saltos (los vecinos de mis vecinos), para ello se analizan cada una de las entradas de la Tabla de Enlaces que corresponden a los vecinos directos, es decir, se analizan los mensajes recibidos de '2.3.1' y '2.1.1', que se acaban de añadir a la tabla a un salto de distancia, y que poseerán vecinos situados a dos saltos del nodo en estudio.

Uno de los vecinos del vecino del nodo en estudio, '2.3.1.1', será el propio nodo, por lo que se ignora. Como ninguno de ellos es a la vez vecino directo del conmutador en estudio, se continúa con el algoritmo y el resto de vecinos se añade a la Tabla de Reenvío con distancia dos, indicando el vecino origen del mensaje como siguiente salto. En la primera entrada de la Tabla de Enlaces, correspondiente al nodo '2.3.1', se descubre que tiene dos vecinos, '2.3' y '1.1.1.3' (distintos del propio conmutador '2.3.1'). En la segunda entrada, correspondiente al nodo '2.1.1' se descubre que tiene el vecino '2.1'. Ambos nodos descubiertos se incorporan a la Tabla de Reenvío con distancia dos.

Tabla 32. Tabla de Reenvío de un *conmutador* D-TRE1 tras el paso 2 del algoritmo de actualización topológica (para el *conmutador* '2.3.1.1' de la red de la Ilustración 40)

Nodo	Siguiente salto	Distancia
2.3.1	2.3.1	1
2.1.1	2.1.1	1
2.3	2.3.1	2
1.1.1.3	2.3.1	2
2.1	2.1.1	2

Finalmente, se incluyen los vecinos a tres saltos. Se analizan cada una de las entradas de la Tabla de Enlaces que corresponden a los vecinos de los vecinos, es decir, se analizan los mensajes difundidos por los nodos situados a dos saltos, que se acaban de añadir a la tabla con distancia dos, y cuyos vecinos estarán situados a tres saltos.

Para el caso del nodo '2.3.1.1', se estudia si se ha recibido mensajes de los nodos incluidos en el campo "Vecinos" de la Tabla de Enlaces para los vecinos directos (que se acaban de añadir a la Tabla de Reenvío)y, por lo tanto, tienen su entrada correspondiente en la Tabla de Enlaces.

En la entrada correspondiente al vecino directo '2.3.1' se incluyen como vecinos los nodos '2.3.1.1', '2.3' y '1.1.1.3'. El primero es el propio nodo, por lo que se ignora. Los dos últimos no son el propio nodo, ni vecinos directos, por lo que se busca en la Tabla de Enlaces si existe alguna entrada para ambos, pero como no se localiza ninguna entrada ('2.3.1' no reenvía mensajes ya que es un nodo de mayor grado), se termina el proceso con el nodo '2.3.1'.

En la entrada correspondiente al vecino directo '2.1.1' se incluyen como vecinos los nodos '2.3.1.1' y '2.1'. El primero es el propio nodo, por lo que se ignora. El segundo ni es el propio nodo, ni un vecino directo, por lo que se busca en la Tabla de Enlaces si existe alguna entrada para éste. En este caso sí existe una entrada para el nodo '2.1', por lo que se procesa la información referente a sus vecinos incluida en la Tabla de Enlaces.

Los vecinos de '2.1' son los nodos '2.1.1', '2' y '1.2'. El primero de ellos es un vecino del conmutador en estudio, por lo que se ignora (está ya incluido en la Tabla de Reenvío con distancia igual a uno). Los dos últimos no están incluidos en la Tabla de Reenvío y se incorporan con distancia igual a tres. El "siguiente salto" es el mismo que para alcanzar el '2.1', que según se observa en la Tabla de Reenvío es el '2.1.1'. La Tabla de Reenvío *definitiva* se muestra en la Tabla 33.

Tabla 33. Tabla de Reenvío para el nodo '2.3.1.1' de la Ilustración 40

Nodo	Siguiente salto	Distancia
2.3.1	2.3.1	1
2.1.1	2.1.1	1
2.3	2.3.1	2
1.1.1.3	2.3.1	2
2.1	2.1.1	2
2	2.1.1	3
1.2	2.1.1	3

4.5.7.3 Análisis de prestaciones

El rendimiento de D-TRE1 se ha evaluado midiendo la longitud media del camino, computada como número de saltos de cada nodo origen hasta los posibles destinos; el coste medio asociado a la decisión de elegir el puerto por el que se encamina una trama, medido como el número de entradas consultadas y comparadas de la Tabla de Reenvío que tienen que realizar los nodos que forman el camino; y el throughput relativo, calculado como el número de flujos unitarios que transcurren por el enlace más ocupado de la red, relativo a un protocolo de tipo *Shortest Path*.

El modelo de tráfico empleado para el análisis de D-TRE1 es el mismo que el utilizado en el análisis del rendimiento de TRE+, se modelan flujos unitarios desde cada nodo a los demás nodos. De esta manera es posible comparar el rendimiento de D-TRE1 con TRE+ y *Shortest Path*.

Para evaluar el rendimiento de D-TRE1 se han simulado distintos tipos de topologías de redes irregulares generadas con la herramienta BRITE, con distintos grados medios (4 y 6) y número de nodos (64, 128, 256, 512 y 1024). Los modelos generados son de escala libre, "Scale-free", (modelos Barabasi-Albert que siguen una distribución exponencial) y aleatorias (modelos Waxman). Para eliminar la dependencia con una raíz concreta elegida, se realizan distintas iteraciones por topología, estableciendo una raíz distinta en cada iteración. Los resultados que se muestran son las medias de las iteraciones por cada topología y las medias de las topologías.

El rendimiento de D-TRE1 en las topologías de tipo Barabasi y Waxman se muestra en la Tabla 34 y Tabla 35, respectivamente.

Tabla 34. Rendimiento comparativo de D-TRE1 en topologías Barabasi

Topología		Longitud media del camino (nº saltos)			Coste en decisión de reenvío (nº consultas)			Throughput (% de ShP)		Ratio Th.
Nodos	Grado medio	TRE+	D-TRE1	ShP	TRE+	D-TRE1	ShP	TRE+	D-TRE1	DTRE1 / TRE+
64	4	2,88	2,86	2,81	55	71	177	92,4	96,0	1,04
	6	2,43	2,41	2,39	72	92	150	96,2	101,1	1,05
128	4	3,22	3,19	3,11	93	128	395	87,5	91,4	1,04
	6	2,77	2,73	2,69	124	177	342	85,8	91,1	1,06
256	4	3,66	3,61	3,48	140	205	887	71,8	78,2	1,09
	6	3,06	3,02	2,95	205	317	752	76,8	84,8	1,10
512	4	3,99	3,94	3,77	222	339	1926	69,57	75,6	1,09
	6	3,43	3,37	3,24	302	517	1656	65,6	77,6	1,18
1024	4	4,40	4,34	4,15	307	471	5268	57,9	70,9	1,22
	6	3,74	3,66	3,51	425	784	3590	51,1	60,4	1,18

Tabla 35. Rendimiento comparativo de D-TRE1 en topologías Waxman

Topología		Longitud media del camino (nº saltos)			Coste en decisión de reenvío (nº consultas)			Throughput (% de ShP)		Ratio Th.
Nodos	Grado medio	TRE+	D-TRE1	ShP	TRE+	D-TRE1	ShP	TRE+	D-TRE1	DTRE1 / TRE+
64	4	3,16	3,09	2,99	48	68	188	76,1	86,2	1,13
	6	2,59	2,55	2,50	64	92	157	87,5	99,0	1,13
128	4	3,85	3,75	3,52	67	102	447	58,2	69,7	1,20
	6	3,04	2,96	2,86	100	166	363	63,6	78,5	1,23
256	4	4,52	4,40	4,02	89	144	1025	39,8	49,8	1,25
	6	3,62	3,48	3,26	136	259	831	43,2	59,1	1,37
512	4	5,16	5,03	4,46	116	187	2279	29,1	36,4	1,25
	6	4,14	3,98	3,62	181	357	1849	30,3	43,2	1,43
1024	4	5,80	5,63	4,92	147	239	5033	22,6	27,7	1,23
	6	4,59	4,40	3,96	235	488	4051	25,3	36,1	1,43

Se observa como D-TRE1 tiene un rendimiento relativo frente a TRE+ entre 1,13 y 1,43 veces para topologías Waxman y entre 1,04 y 1,22 veces para topologías Barabasi. Se obtiene un mayor beneficio cuanto mayor es el número de nodos y mayor es el grado medio de la red al disponer de mayor número de atajos accesibles.

La longitud de los caminos se acorta en D-TRE1 respecto a TRE+, aproximándose a la longitud de *Shortest Path* y situándose a valores de entre un 1% y un 4% en redes Barabasi y entre un 2 y un 12% en redes Waxman.

D-TRE1 presenta menor complejidad computacional que Shortest Path. D-TRE1 necesita entre un 40% y un 95% menos de consultas a la tabla para comparar distancias y decidir el puerto por el que se reenvía la trama. El 40% de mejora se obtiene en redes de menor número de nodos y mayor

grado medio, mientras que el 95% de mejora se obtiene en redes de mayor número de nodos y menor grado medio.

4.5.8 El protocolo D-TRE2

En el protocolo D-TRE2, el alcance de la información que adquieren los conmutadores depende del nivel en el que estén situados. Se considera que es en los niveles superiores donde más nodos existen y, por lo tanto, donde hay más probabilidad de utilizar atajos. En este protocolo se dota de autonomía al conmutador, ya que su alcance depende de un parámetro interno del mismo, el nivel del árbol en el que está situado, sin condicionar su funcionamiento por un parámetro externo como ocurre en D-TRE1 con el grado medio de la red.

La información topológica que poseerán los conmutadores D-TRE2 tendrá un alcance, medido en número de saltos, que variará según su situación en el árbol, siendo mayor en los niveles superiores y menor en los niveles inferiores, pero siempre aplicando como mínimo un alcance igual a uno (caso similar a TRE), ya que son conmutadores de la arquitectura TRE. Los dos niveles superiores (1º y 2º) tienen un alcance igual a tres, los dos intermedios (3º y 4º) igual a dos, y los dos niveles inferiores (5º y 6º) igual a uno.

A continuación se analiza el comportamiento del protocolo a partir de la red de la Ilustración 46.

Cada nodo analiza su dirección HLMAC para conocer el nivel del árbol en el que está situado. Como se ha explicado en el apartado 4.3.1, una dirección HLMAC está compuesta por 6 octetos en cada uno de los cuáles se codifica un nivel del árbol del expansión. El nivel de un nodo se corresponde con el número de octetos no nulos de su dirección.

Por ejemplo, el nodo '1', con HLMAC '1.0.0.0.0.0' pertenece al nivel 1º (conectado a la raíz). Por tanto, le corresponde un alcance de tres saltos (tendrá conocimiento de los nodos situados a tres saltos de distancia). En relación a la red de la Ilustración 46 descubrirá, además de la práctica totalidad de los nodos descendentes por sus ramas, los nodos '2.1', '2.2', '2.3' y '2.1.1' (gracias al atajo que le proporciona el enlace [1 – 2.1]), y el nodo '2.2.1.1' (gracias al atajo [2.1.1 – 2.2.1.1] que es precisamente el tercer salto).

Por su parte, el nodo '2.1.1', con HLMAC '2.1.1.0.0.0', pertenece al nivel 3º y tendrá alcance dos, por lo que tendrá conocimiento de los nodos situados a dos saltos de distancia. Se muestran dentro del trazo rojo todos los nodos conocidos por '2.1.1'.

Por el último, el nodo '1.1.1.2.1.1' pertenece al nivel 6º, tendrá alcance de solo un salto (sus vecinos). En la ilustración se muestran dentro del trazo verde.

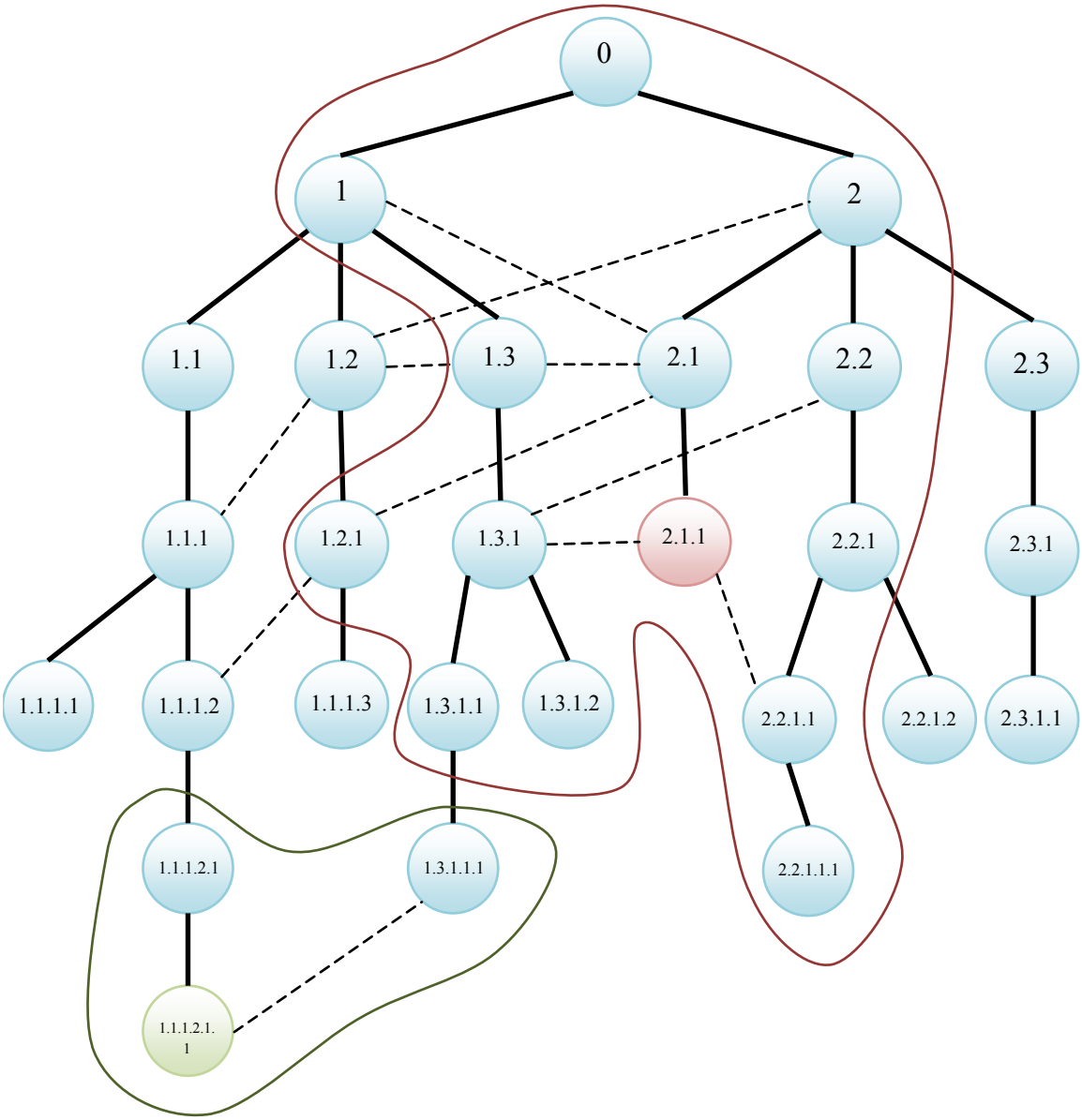


Ilustración 46. Alcance de la información topológica de los nodos D-TRE2

4.5.8.1 Intercambio de mensajes

El procedimiento de intercambio de mensajes topológicos se realiza mediante el envío de un mensaje de control D-TLC con la estructura descrita en el apartado 4.5.3

El tratamiento (preparación y reenvío) de los mensajes dependerá del valor del campo TTL con el que se controla el alcance de la información de vecindad codificada y dependerá del nivel del árbol al que pertenezca dicho nodo:

En los nodos del nivel 1º (conectados a la raíz):

- Si el nodo genera un mensaje: establece el valor TTL=2 y envía el mensaje hacia todos sus vecinos, sean del nivel que sean.
- Si el nodo recibe un mensaje: actualiza su Tabla de Reenvío, decrementa el valor del TTL y reenvía el mensaje hacia todos sus vecinos, sean del nivel que sean.

En los nodos de segundo nivel:

- Si el nodo genera un mensaje: establece el valor TTL=2 y reenvía el mensaje hacia todos sus vecinos, sean del nivel que sean.
- Si el nodo recibe un mensaje: actualiza su Tabla de Reenvío, decrementa el valor del TTL y reenvía el mensaje hacia sus vecinos de niveles superiores (nivel 1º) y del mismo nivel (nivel 2º), si el valor de TTL es distinto de cero.

En los nodos de tercer nivel:

- Si el nodo genera un mensaje: establece el valor TTL=2 y envía el mensaje hacia sus vecinos del mismo nivel (3º) o nivel superior (2º), después decrementa el valor del TTL (TTL=1) y envía el mensaje a sus vecinos de nivel inferior (4º).
- Si el nodo recibe un mensaje: actualiza su Tabla de Reenvío, decrementa el valor del TTL y reenvía el mensaje hacia sus vecinos de niveles superiores (nivel 2º), si el valor de TTL es distinto de cero.

En los nodos de cuarto nivel:

- Si el nodo genera un mensaje: establece el valor TTL=2 y envía el mensaje hacia sus vecinos nivel superior (3º), después decrementa el valor del TTL (TTL=1) y envía el mensaje a sus vecinos del mismo nivel (4º).
- Si el nodo recibe un mensaje (no debería producirse): actualiza su Tabla de Reenvío, decrementa el valor del TTL y reenvía el mensaje hacia sus vecinos de niveles superiores (nivel 3º), si el valor de TTL es distinto de cero.

En los nodos de quinto nivel:

- Si el nodo genera un mensaje: establece el valor TTL=1 y envía el mensaje hacia sus vecinos nivel superior (4º).
- Si el nodo recibe un mensaje (no le debería llegar): actualiza su Tabla de Reenvío y descarta el mensaje.

Por último, los nodos del nivel 6º no generan mensajes ni deberían recibirlos. En caso de recepción de un mensaje de control actualizaría su tabla y descartaría el mensaje.

4.5.8.2 Ejemplo de operación

En el siguiente ejemplo se detalla el proceso de gestión de la información topológica para el nodo '2.1.1' de la Ilustración 46. Se trata de un nodo del nivel 3º, por lo tanto, deberá tener conocimiento de la existencia de los nodos situados a dos saltos de distancia. En primer lugar se analizan los mensajes recibidos. Posteriormente se crea la Tabla de Enlaces a partir de la información que contienen dichos mensajes. Por último se detalla cómo se genera la Tabla de Reenvío.

Recibe y procesa los mensajes generados por sus vecinos '2.1', '1.3.1' y '2.2.1.1':

El nodo '2.1', como todo nodo de nivel 2º, envía su mensaje D-TLC a todos sus vecinos, independientemente de sus niveles, indicando que tiene a los '2', '2.1.1', '1', '1.3' y '1.3.1' a un salto.

Nodo origen	TTL	Vecinos				
2.1	TTL=2	2	2.1.1	1	1.3	1.3.1

Ilustración 47. Mensaje generado por el nodo '2.1' de la Ilustración 46

El nodo '1.3.1' es un nodo de nivel 3º, e informa que sus vecinos son '1.3', '1.3.1.1', '1.3.1.2', '2.1.1' y '2.2'.

Nodo origen	TTL	Vecinos				
1.3.1	TTL=2	1.3	1.3.1.1	1.3.1.2	2.1.1	2.2

Ilustración 48. Mensaje generado por el nodo '1.3.1' de la Ilustración 46

Por último, el nodo '2.2.1.1' es un nodo de nivel 4º, informa que sus vecinos son '2.2.1', '2.2.1.1.1' y '2.1.1'.

Nodo origen	TTL	Vecinos		
2.2.1.1	TTL=2	2.2.1	2.2.1.1.1	2.1.1

Ilustración 49. Mensaje generado por el nodo '2.2.1.1' de la Ilustración 46

También debe recibir y procesar los mensajes reenviados por sus vecinos, en este caso, al tratarse de un nodo de tercer nivel recibiría mensajes reenviados de nodos de nivel 2º, 3º o 4º. Los nodos de nivel 2º solo reenvían a iguales o superiores (no es el caso), los de nivel 3º solo a superiores (tampoco aplica) y los de 4º nivel no reenvían mensajes. En definitiva, el nodo '2.1.1' no debería recibir mensajes reenviados por sus vecinos.

La Tabla de Enlaces y la Tabla de Reenvío se configuran de la misma forma que en el caso de D-TRE1. Con los mensajes recibidos se configura la Tabla de enlaces siguiente:

Tabla 36. Tabla de Enlaces del conmutador '2.1.1' para la red de la Ilustración 46

Nodo origen	Puerto recibido	Nº Sec.	Vecinos				
2.1	2.1	1	2	2.1.1	1	1.3	1.3.1
1.3.1	1.3.1	1	1.3	1.3.1.1	1.3.1.2	2.1.1	2.2
2.2.1.1	2.2.1.1	1	2.2.1	2.2.1.1.1	2.1.1		

Por su parte, la Tabla de Reenvío, se construye incluyendo primero a los vecinos para conformar la Tabla de Reenvío inicial mostrada en la Tabla 37.

Tabla 37. Tabla de Reenvío de un conmutador D-TRE2 tras el paso 1 del algoritmo de actualización topológica (para el conmutador '2.1.1' la red de la Ilustración 46)

Nodo	Siguiente salto	Distancia
2.1	2.1	1
1.3.1	1.3.1	1
2.2.1.1	2.2.1.1	1

Después se incluyen los nodos situados a dos saltos de distancia. Para ello se analizan las entradas de la Tabla de Enlaces que corresponden a los vecinos directos, es decir, se analizan los mensajes recibidos de '2.1', '1.3.1' y '2.2.1.1', que se acaban de añadir a la tabla a un salto de distancia, y poseerán vecinos situados a dos saltos del nodo en estudio. Como ninguno de ellos es un vecino directo, se añaden todos como vecinos a dos saltos tomando el vecino correspondiente como siguiente salto. En la primera entrada de la Tabla de Enlaces, correspondiente al nodo '2.1', se descubre que tiene cuatro vecinos, '2', '1', '1.3' y '1.3.1' (distintos del propio conmutador '2.1.1'). En la segunda entrada, correspondiente al nodo '1.3.1', se descubre que tiene otros cuatro vecinos, '1.3', '1.3.1.1', '1.3.1.2' y '2.2' y, finalmente, en la última entrada, correspondiente al nodo '2.2.1.1', se descubre que tiene dos vecinos, '2.2.1' y '2.2.1.1.1' (obviamente también el propio conmutador '2.1.1' que se ignora).

Tabla 38. Tabla de Reenvío de un conmutador D-TRE2 tras el paso 2 del algoritmo de actualización topológica (para el conmutador '2.1.1' de la red de la Ilustración 46)

Nodo	Siguiente salto	Distancia
2.1	2.1	1
1.3.1	1.3.1	1
2.2.1.1	2.2.1.1	1
2	2.1	2
1.3	2.1	2
1	2.1	2
1.3.1	2.1	2
1.3	1.3.1	2
1.3.1.1	1.3.1	2
1.3.1.2	1.3.1	2
2.2	1.3.1	2
2.2.1	2.2.1.1	2
2.2.1.1.1	2.2.1.1	2

Por último, se incluyen los vecinos a tres saltos de distancia. Como se trata de un nodo de nivel 3º, no recibe información de nodos situados a tres saltos, es decir, vecinos anunciados por los vecinos de los vecinos.

Si fuera de nivel 1º ó 2º, habría que analizar los mensajes reenviados por los vecinos, según lo estipulado en el Algoritmo de Actualización Topológica para la Tabla de Reenvío, añadiendo en esta tabla con distancia tres, los nuevos vecinos aprendidos (siempre que no sean vecinos ya conocidos, que estarán añadidos ya con distancia uno, si son directos, o con distancia dos, si son vecinos de vecinos).

4.5.8.3 Análisis de prestaciones

El rendimiento de D-TRE2 se ha evaluado (al igual que para D-TRE1 en la sección anterior) midiendo la longitud media del camino, computada como el número de saltos de cada nodo origen hasta los posibles destinos; el coste medio asociado a la decisión de elegir el puerto por el que se encamina una trama, medido como el número de entradas consultadas y comparadas de la Tabla de Reenvío que tienen que realizar los nodos que forman el camino; y el throughput relativo, computado como el número de flujos unitarios que discurren por el enlace más ocupado de la red, relativo a un protocolo de tipo *Shortest Path*.

El modelo de tráfico empleado para el análisis de D-TRE2 es el utilizado en el análisis del rendimiento de TRE+, se modelan flujos unitarios desde cada nodo a los nodos restantes. De esta manera es posible comparar el rendimiento de D-TRE2 con TRE+ y *Shortest Path*.

Para evaluar el rendimiento de D-TRE2 se han simulado distintos tipos de topologías de redes irregulares generadas con la herramienta BRITE, con distintos grados medios (4 y 6) y número de nodos (64, 128, 256, 512 y 1024). Los modelos generados son de escala libre, "*scale-free*", (modelos Barabasi-Albert que siguen una distribución exponencial) y aleatorias (modelos Waxman). Para eliminar la dependencia con una raíz concreta elegida, se realizan distintas iteraciones por topología, seleccionando una raíz distinta en cada iteración. Los resultados obtenidos corresponden con las medias de las iteraciones por cada topología, y las medias de las topologías. El rendimiento de D-TRE2 en las topologías de Barabasi y Waxman y se muestran en la Tabla 39 y la Tabla 40, respectivamente.

Tabla 39. Rendimiento comparativo de D-TRE2 en topologías Barabasi

Topología		**Longitud media del camino (nº saltos)**			**Coste en decisión de reenvío (nº consultas)**			**Throughput (% de ShP)**		**Ratio Th.**
Nodos	Grado medio	TRE+	D-TRE2	ShP	TRE+	D-TRE2	ShP	TRE+	D-TRE2	DTRE2 / TRE+
64	4	2,88	2.85	2,81	55	62	177	92,4	97,2	1,05
	6	2,43	2.41	2,39	72	79	150	96,2	102,6	1,07
128	4	3,22	3.18	3,11	93	117	395	87,5	91,9	1,05
	6	2,77	2.73	2,69	124	154	342	85,8	91,9	1,07
256	4	3,66	3.58	3,48	140	226	887	71,8	81,9	1,14
	6	3,06	3.00	2,95	205	314	752	76,8	90,2	1,17
512	4	3,99	3.91	3,77	222	409	1926	69,5	76,4	1,10
	6	3,43	3.33	3,24	302	613	1656	65,6	89,7	1,37
1024	4	4,40	4.30	4,15	307	723	5268	57,9	74,7	1,29
	6	3,74	3.59	3,51	425	1130	3590	51,1	73,8	1,44

D-TRE2 tiene un rendimiento relativo a TRE+ de entre un 120% y un 200% para topologías Waxman y entre un 105% y un 144% para topologías Barabasi. Se obtiene mayor beneficio cuanto mayor es el número de nodos y mayor es el grado medio de la red.

La longitud de los caminos se acorta en D-TRE2 respecto a TRE+, aproximándose a la longitud de *Shortest Path* y situándose a valores de entre un 1% y un 3.5% en redes Barabasi y entre un 1,5% y un 12% en redes Waxman.

Tabla 40. Rendimiento comparativo de D-TRE2 en topologías Waxman

Topología		Longitud media del camino (nº saltos)			Coste en decisión de reenvío (nº consultas)			Throughput (% de ShP)		Ratio Th.
Nodos	Grado medio	TRE+	D-TRE2	ShP	TRE+	D-TRE2	ShP	TRE+	D-TRE2	DTRE2/ TRE+
64	4	3,16	3.08	2,99	48	63	188	76,1	91,5	1,20
	6	2,59	2.54	2,50	64	81	157	87,5	105,2	1,20
128	4	3,85	3.73	3,52	67	105	447	58,2	79,7	1,37
	6	3,04	2.94	2,86	100	161	363	63,6	92,1	1,45
256	4	4,52	4.38	4,02	89	166	1025	39,8	61,5	1,55
	6	3,62	3.46	3,26	136	259	831	43,2	79,3	1,84
512	4	5,16	5.01	4,46	116	242	2279	29,1	47,5	1,63
	6	4,14	3.95	3,62	181	357	1849	30,3	61,5	2,03
1024	4	5,80	5.61	4,92	147	383	5033	22,6	37,2	1,65
	6	4,59	4.35	3,96	235	488	4051	25,3	50,4	1,99

D-TRE2 presenta menor complejidad computacional que *Shortest Path*, necesita entre un 47% y un 86% menos de consultas a la tabla para comparar distancias y decidir el puerto por el que se reenvía la trama. El 47% de mejora se obtiene en redes de menor número de nodos y mayor grado medio, mientras que el 92% de mejora se obtiene en redes de mayor número de nodos y menor grado medio.

4.5.9 Análisis de la sobrecarga

El intercambio de mensajes de control D-TLC es necesario para conocer los vecinos de los nodos que están a más de un salto de distancia. Este flujo de mensajes de control repercute en la capacidad de los enlaces de la red, introduciendo una carga adicional al tráfico generado por las tramas de datos. La sobrecarga producida por el tráfico de mensajes de control debe ser la menor posible para que no se vea afectada la capacidad efectiva del enlace. Es importante conocer cuánta cantidad de información se debe introducir en la red para que todos los nodos de la red adquieran el conocimiento topológico estipulado en cada uno de los dos protocolos D-TRE descritos en los secciones anteriores. Para ello se ha implementado un programa en Python, cuyo parámetro de entrada es la red a estudio, que analiza el plano de control y proporciona estadísticas de la sobrecarga y la estabilidad de la red.

El tamaño del mensaje de control D-TLC generado por cada nodo depende del número de vecinos que posea, debido a que incluyen como parte variable las direcciones de los vecinos del nodo origen. Al tamaño fijo (10 octetos) de la cabecera del mensaje de control D-TLC (sección), hay que añadir el tamaño variable del campo 'Datos', 6 octetos por cada vecino. Además, debemos tener en cuenta que una trama Ethernet debe alcanzar un tamaño mínimo que obliga a rellenar el campo de datos hasta ocupar al menos 46 octetos y que éste debe ser múltiplo de 8 octetos.

4.5.9.1 Sobrecarga en D-TRE1

En el caso de D-TRE1, la cantidad de mensajes de control necesarios depende en primer lugar de la relación entre el grado de cada nodo y el grado medio de la red (d) que regirá el comportamiento

de cada nodo y, después, del número concreto de vecinos a distancia un salto y distancia dos saltos de cada nodo.

Las Tablas 41 y 42 muestran los valores de sobrecarga calculados (medidos en número de mensajes de control por enlace de la red) para redes aleatorias de tipo Barabasi y Waxman, respectivamente, con tamaños de entre 64 y 1024 nodos y grado medio de valores 4 y 6. Además del valor medio por enlace, incluyen también los valores máximos y el total de mensajes en la red.

Tabla 41. Sobrecarga por enlace en D-TRE1 (nº mensajes) en redes Barabasi

Nº Nodos	Grado medio	Máximo	Media	Total
64	4	4,0	1,88	470
	6	6,0	2,51	934
128	4	4,0	1,86	940
	6	6,0	2,54	1.920
256	4	4,0	1,84	1.877
	6	6,0	2,55	3.889
512	4	4,0	1,84	3.758
	6	6,0	2,55	7.816
1024	4	4,0	1,86	7.595
	6	6,0	2,52	15.476

Tabla 42. Sobrecarga por enlace en D-TRE1 (nº mensajes) en redes Waxman

Nº Nodos	Grado medio	Máximo	Media	Total
64	4	4,0	1,96	502
	6	6,0	2,55	979
128	4	4,0	1,94	993
	6	6,0	2,62	2.012
256	4	4,0	1,97	2.017
	6	6,0	2,67	4.108
512	4	4,0	1,95	3.991
	6	6,0	2,64	8.112
1024	4	4,0	1,94	7.939
	6	6,0	2,66	16.335

Se observa como las características de sobrecarga por enlace en la red, evaluadas en número de mensajes, dependen exclusivamente del grado medio de la red, independientemente del número de nodos que posea. Sin embargo, la cantidad de información transmitida en cada enlace (medida en número de bytes) no es proporcional al número de mensajes por enlace, ya que el tamaño de los mensajes D-TLC no es fijo, sino que depende de los vecinos que posea cada nodo. Este tráfico (medido en número de bytes) se calcula en el siguiente apartado y se refiere al periodo de envío de los mensajes por cada nodo.

Para transformar la medida anterior en un factor de utilización de la capacidad de los enlaces, debemos incorporar al análisis una nueva variable, el tiempo de periodicidad de los envíos. Se han estudiado cinco posibles periodos de reenvío: 100 ms, 200 ms, 500 ms, 1 s y 2 s. Y los resultados se presentan en las tablas siguientes.

Tabla 43. Sobrecarga por enlace en D-TRE1 (*en Kbps*) en redes Barabasi

Nº Nodos (N)	Grado medio	Sobrecarga por enlace (octetos)	Periodicidad de envío				
			100 ms	200 ms	500 ms	1 s	2s
64	4	168,32	13,15	6,58	2,63	1,32	0,66
	6	250,98	19,61	9,80	3,92	1,96	0,98
128	4	181,38	14,17	7,09	2,83	1,42	0,71
	6	276,95	21,64	10,82	4,33	2,16	1,08
256	4	189,02	14,77	7,38	2,95	1,48	0,74
	6	301,92	23,59	11,79	4,72	2,36	1,18
512	4	204,65	15,99	7,99	3,20	1,60	0,80
	6	316,14	24,70	12,35	4,94	2,47	1,23
1024	4	212,60	16,61	8,30	3,32	1,66	0,83
	6	336,36	26,28	13,14	5,26	2,63	1,31

Tabla 44. Sobrecarga por enlace en D-TRE1 (*en Kbps*) en redes Waxman

Nº Nodos (N)	Grado medio	Sobrecarga por enlace (octetos)	Periodicidad de envío				
			100 ms	200 ms	500 ms	1 s	2s
64	4	154,02	12,03	6,02	2,41	1,20	0,60
	6	220,25	17,21	8,60	3,44	1,72	0,86
128	4	152,35	11,90	5,95	2,38	1,19	0,60
	6	230,97	18,04	9,02	3,61	1,80	0,90
256	4	155,84	12,18	6,09	2,44	1,22	0,61
	6	234,61	18,33	9,16	3,67	1,83	0,92
512	4	154,01	12,03	6,02	2,41	1,20	0,60
	6	233,48	18,24	9,12	3,65	1,82	0,91
1024	4	153,41	11,99	5,99	2,40	1,20	0,60
	6	237,73	18,57	9,29	3,71	1,86	0,93

La sobrecarga es mayor en redes de grado medio mayor, ya que hay más cantidad de nodos con más de 6 vecinos. La sobrecarga por enlace apenas varía con el incremento de número de nodos en la red, debido a que aunque aumente el número de nodos, el grado medio no varía.

De los resultados obtenidos se concluye que la sobrecarga introducida por el protocolo D-TRE1 es mínima en cualquier tipo de red, pudiendo ser considerada despreciable. Referida a un sistema *Fast Ethernet*, con velocidad de enlace de 100 Mbps, correspondería con una ocupación máxima de 0,026% en Barabasi y de 0,018% en redes Waxman.

4.5.9.2 Sobrecarga en D-TRE2

En el caso de D-TRE2, la cantidad de mensajes de control necesarios depende en primer lugar del nivel que ocupa cada nodo en el árbol de expansión además del número de vecinos a distancia un salto y distancia dos saltos.

La Tabla 45 y la Tabla 46 muestran los valores de sobrecarga calculados (medidos en número de mensajes de control por enlace de la red) para redes aleatorias de tipo Barabasi y Waxman

respectivamente con tamaños de entre 64 y 1024 nodos y grado medio de valor 4 y 6. Además del valor medio por enlace, incluyen también los valores máximos y el total de mensajes en la red.

Tabla 45. Sobrecarga por enlace en D-TRE2 (nº mensajes) en redes Barabasi

Nº Nodos	Grado medio	Máximo	Media	Total
64	4	19,9	4,98	1.245
	6	22,3	7,45	2.771
128	4	26,0	5,00	2.530
	6	36,4	7,99	6.042
256	4	39,4	4,86	4.951
	6	49,8	7,91	12.053
512	4	66,6	5,12	10.447
	6	68,2	7,63	23.361
1024	4	88,0	5,41	22.136
	6	98,0	7,84	48.066

Tabla 46. Sobrecarga por enlace en D-TRE2 (nº mensajes) en redes Waxman

Nº Nodos	Grado medio	Máximo	Media	Total
64	4	12,4	3,47	888
	6	16,8	6,17	2.370
128	4	14,8	2,82	1.444
	6	18,8	5,41	4.152
256	4	15,0	2,46	2.524
	6	20,8	4,53	6.952
512	4	18,2	1,80	3.683
	6	25,0	3,68	11.290
1024	4	22,0	1,83	7.484
	6	31,5	2,88	17.712

En este caso, las características de sobrecarga por enlace en la red, evaluadas en número de mensajes, no dependen exclusivamente del grado medio de la red, como ocurría en D-TRE1, sino también del número de nodos de la red.

Además, la cantidad de información transmitida en cada enlace (medida en octetos) no es proporcional al número de mensajes por enlace, ya que el tamaño de los mensajes D-TLC no es fijo, sino que depende de los vecinos que posea cada nodo además de la periodicidad de envío de los mensajes de control. Los resultados se muestran en las tablas siguientes.

Tabla 47. Sobrecarga por enlace en D-TRE2 (*en Kbps*) en redes Barabasi

Nº Nodos (N)	Grado medio	Sobrecarga por enlace (octetos)	Periodicidad de envío				
			100 ms	200 ms	500 ms	1 s	2 s
64	4	419,75	32,79	16,40	6,56	3,28	1,64
	6	695,75	54,36	27,18	10,87	5,44	2,72
128	4	447,88	34,99	17,50	7,00	3,50	1,75
	6	797,79	62,33	31,16	12,47	6,23	3,12
256	4	453,52	35,43	17,72	7,09	3,54	1,77
	6	841,92	65,78	32,89	13,16	6,58	3,29
512	4	498,08	38,91	19,46	7,78	3,89	1,95
	6	851,00	66,48	33,24	13,30	6,65	3,32
1024	4	556,45	43,47	21,74	8,69	4,35	2,17
	6	919,91	71,87	35,93	14,37	7,19	3,59

Tabla 48. Sobrecarga por enlace en D-TRE2 (*en Kbps*) en redes Waxman

Nº Nodos (N)	Grado medio	Sobrecarga por enlace (octetos)	Periodicidad de envío				
			100 ms	200 ms	500 ms	1 s	2 s
64	4	271,31	21,20	10,60	4,24	2,12	1,06
	6	541,04	42,27	21,13	8,45	4,23	2,11
128	4	223,37	17,45	8,73	3,49	1,75	0,87
	6	485,85	37,96	18,98	7,59	3,80	1,90
256	4	199,61	15,59	7,80	3,12	1,56	0,78
	6	413,97	32,34	16,17	6,47	3,23	1,62
512	4	146,00	11,41	5,70	2,28	1,14	0,57
	6	343,44	26,83	13,42	5,37	2,68	1,34
1024	4	151,02	11,80	5,90	2,36	1,18	0,59
	6	275,42	21,52	10,76	4,30	2,15	1,08

Al igual que en D-TRE1, la sobrecarga puede despreciarse. En las redes Barabasi, la mayor sobrecarga es de 71,87 Kbps (para el caso de redes con N=1024 nodos, grado medio d=6 y periodicidad de envío T=100 ms), mientras que en Waxman, es de 42,27 Kbps (para el caso de redes con 64 nodos, grado medio d=6 y periodicidad de envío T=100 ms). Referida a un sistema *Fast Ethernet*, con velocidad de enlace de 100 Mbps, correspondería con una ocupación máxima de 0,07% en ese tipo de redes Barabasi, y 0,041% en redes Waxman.

4.5.10 Análisis de estabilidad

Cuando un nodo D-TRE recibe un mensaje de control debe realizar dos tareas:

1. Almacenar la información topológica, mediante el Algoritmo de Actualización Topológica de la Tabla de Enlaces y la Tabla de Reenvío, descrito apartados anteriores, cuyo fin es realizar el encaminamiento de las tramas de datos habiendo alcanzado una estabilidad inicial.

2. Analizar posibles modificaciones de la red considerando posibles fallos y/o cambios topológicos, cuyo fin es garantizar la estabilidad en el protocolo.

El protocolo D-TRE utiliza un enrutamiento tipo *estado de enlace*, que requiere el reenvío limitado de mensajes para alcanzar una estabilidad inicial. Una vez intercambiados ese tráfico y obtenida la estabilidad inicial, ésta se debe mantener, analizando los sucesivos mensajes recibidos con el fin de descubrir si se han producido cambios en la topología o fallos en los elementos de la red que puedan provocar inestabilidades.

4.5.10.1 Convergencia inicial

El tráfico (en octetos) recibido en el primer periodo de intercambio de mensajes es el necesario para conseguir la estabilidad inicial de las tablas de reenvío, es decir, contiene la información imprescindible para que un nodo realice el encaminamiento de las tramas de datos empleando el protocolo D-TRE. Por ejemplo, en una red de grado medio 4, con 128 nodos, se han de transmitir una media de 447,88 octetos por enlace, como se puede observar de la Tabla 47. En una red ideal, sin fallos ni cambios topológicos, este primer intercambio sería el único necesario para realizar el encaminamiento.

El nodo receptor de estos mensajes de control actualiza la Tabla de Enlaces, añadiendo a cada entrada una marca de tiempo. La primera marca de tiempo indica el instante en el que se ha registrado la nueva entrada en la tabla. Esta marca de tiempo se actualizará con cada recepción y es necesaria para conocer cuándo se debe reaccionar ante posibles fallos.

4.5.10.2 Cambios de topología (reconfiguración)

Toda recepción de un mensaje de control implica la modificación de una entrada en la Tabla de Enlaces. Incluso si un nodo origen anuncia los mismos vecinos que aparecen en la tabla, al menos se modifica el número de secuencia de la entrada en la tabla, estableciendo el contenido en el mensaje. Esto es así para que un mensaje con un número de secuencia mayor y, por lo tanto, con información topológica más actualizada, sea tenido en cuenta en lugar de un mensaje con número de secuencia menor que ha llegado después (por ejemplo por lentitud en la gestión de un nodo).

De la misma manera, con toda recepción de un mensaje siempre se actualiza la marca de tiempo existente en la entrada de la tabla a la que hace referencia. La marca de tiempo es información local, e indica el último instante en el que se ha actualizado una entrada de la tabla y, por tanto, la última recepción de un mensaje de una dirección origen dada. Este temporizador se emplea para prevenir inestabilidades en la red ante fallos en la misma.

Sin embargo, pueden producirse cambios en la topología activa de la red debido a sucesos como la activación/caída de enlaces y/o nodos, bien sean programados por el administrador de la red o fortuitos. Como resultado del cambio, el nodo comenzará a recibir mensajes de control con nueva información topológica (nuevos vecinos, nuevos nodos, etc.) o, en el peor de los casos, si el problema afecta al árbol de expansión el propio protocolo RSTP base inhabilitará el árbol para reconstruirlo, produciendo a su vez el descarte de las direcciones jerárquicas y el cese del reenvío hasta que todo el sistema se reconfigura por completo (árbol y direcciones).

4.5.10.2.1 Cambios en los enlaces ajenos al árbol

El protocolo se adapta ante los cambios topológicos que se deben a la incorporación o eliminación voluntaria de enlaces considerados como atajos. En este caso la reacción se realiza desde el plano del nodo, el cual tiene que ser autónomo para considerar los cambios en los atajos.

- Si un nodo recibe un mensaje con un vecino nuevo, conectado por atajo: significa que existe un nuevo enlace en la red que puede ser utilizado como atajo. En este caso se modifica la Tabla de Enlaces incorporando el nuevo vecino en la entrada correspondiente al nodo origen y se actualiza la Tabla de Reenvío, considerando el nodo para el cálculo de distancias y el encaminamiento de las tramas.

  Por ejemplo, en la Ilustración 50 se incorpora un nuevo atajo entre los nodos '2.1' y '1.2.1'.

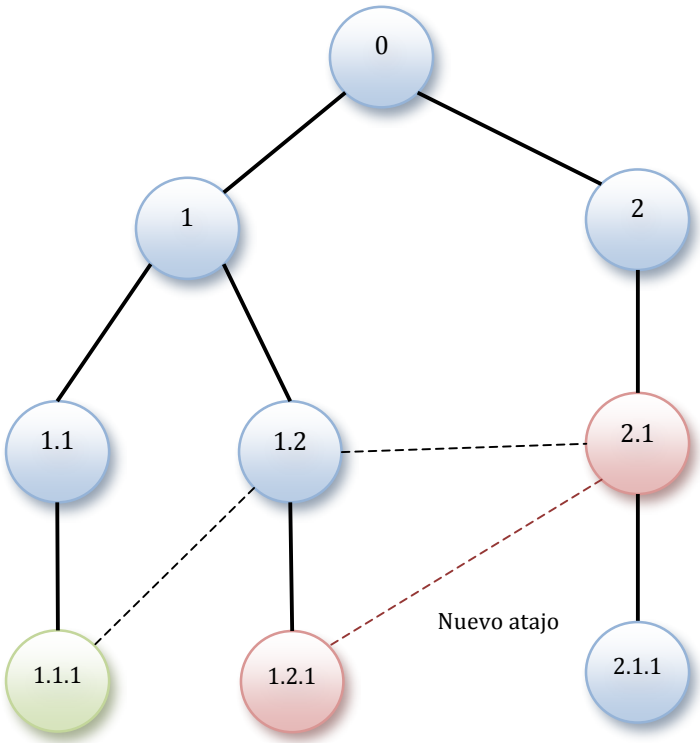


Ilustración 50. Topología de árbol de expansión, después de añadir un nuevo atajo "2.1-1.2.1"

El nodo '1.1.1' dispone de la siguiente Tabla de Enlaces, formada por los mensajes recibidos de los distintos "nodo origen", antes de la incorporación del nuevo atajo entre '1.2.1' y '2.1'.

Tabla 49. Tabla de Enlaces antes de añadir un atajo (del nodo '1.1.1')

Nodo origen	Puerto RX	Nº Secuencia	Vecinos			
1.1	1.1	1	1	1.1.1		
1	1.1	1	0	1.1	1.2	
1.2	1.2	1	1	1.2.1	1.1.1	2.1
1	1.2	1	0	1.1	1.2	
1.2.1	1.2	1	1.2			
2.1	1.2	1	2	1.2	2.1.1	

En el siguiente intercambio de mensajes se añade la información referida al nuevo enlace. Los vecinos de las 4 primeras entradas permanecerán sin cambios porque no se produce modificación en los mensajes recibidos posteriormente. Sin embargo, las últimas dos entradas variarán, ya que '1.1.1' recibe los siguientes mensajes con origen en '1.2.1' y '2.1' (reenviados a través de '1.2'), incluyendo la incorporación de nuevos vecinos:

Nodo origen	Nº secuencia	Vecinos			
1.2.1	Seq=2	1.2	**2.1**		
2.1	Seq=2	2	1.2	2.1.1	**1.2.1**

Ilustración 51. Nuevos vecinos recibidos tras añadir un nuevo atajo (nodo '1.1.1')

Cuando el nodo '1.1.1' recibe el mensaje del nodo '1.2.1':

1. Compara la entrada de su Tabla de Enlaces referida al nodo origen '1.2.1', observando que se anuncia un nuevo nodo vecino, '2.1', respecto a la información existente en la tabla.

2. Comprueba que se trata de un enlace correspondiente a un atajo, y no al árbol (ya que implicaría realizar de nuevo la fase de construcción del árbol y asignación de HLMAC). Es un atajo porque el nuevo vecino anunciado, '2.1', y el nodo origen, '1.2.1', no comparten raíz en su dirección HLMAC.

3. Modifica la Tabla de Enlaces añadiendo el nuevo vecino en esta entrada y actualizando el número de secuencia. Igualmente se actualiza la marca de tiempo para esa entrada.

De igual manera se procede para el mensaje del nodo '2.1', en el que anuncia un nuevo vecino, '1.2.1', resultando la Tabla de Enlaces definitiva:

Tabla 50. Tabla de enlaces tras añadir un nuevo atajo (para el *conmutador* '1.1.1')

Nodo origen	Puerto RX	Nº Secuencia	Vecinos			
1.1	1.1	2	1	1.1.1		
1	1.1	2	0	1.1	1.2	
1.2	1.2	2	1	1.2.1	1.1.1	2.1
1	1.2	2	0	1.1	1.2	
1.2.1	1.2	2	1.2	**2.1**		
2.1	1.2	2	2	1.2	2.1.1	**1.2.1**

- Si un nodo recibe un mensaje nuevo a través de un atajo: De la misma manera que si un nodo anuncia como vecino a otro nodo a través de un atajo (como se ha explicado en el punto anterior), también se puede reconocer que un enlace "atajo" ha comenzado a funcionar si se recibe un mensaje nuevo reenviado a través de dicho atajo.

Por ejemplo, en la Ilustración 52, el nodo '2.1.1' recibirá un nuevo mensaje, con origen en '1.2.1' y reenviado por '2.1'.

1.2.1	Seq=2	1.2	2.1		

Ilustración 52. Nuevo mensaje recibido tras añadir un nuevo atajo

El nodo '2.1.1' al recibirlo descubre que existe un nuevo enlace, y que es un atajo porque el origen es '1.2.1', es reenviado por '2.1' y ambos no comparten raíz en sus direcciones HLMAC. Posteriormente, el nodo '2.1.1' actualiza sus tablas, asumiendo que el nodo '1.2.1' es alcanzable con dos saltos, en lugar de tres.

- Si un nodo recibe un mensaje con un vecino menos, conectado por atajo: Significa que un atajo que se usaba con anterioridad ha dejado de existir. En este caso se modifica la Tabla de Enlaces eliminando este nodo vecino de la entrada correspondiente al nodo origen y se actualiza la Tabla de Reenvío, para no considerar accesible el nodo conectado mediante el enlace caído.

- Si un nodo no recibe un mensaje de un vecino a través de un atajo: De la misma manera que si un nodo no es anunciado como vecino de otro a través de un atajo (como se ha explicado en el punto 3), también se puede reconocer que un enlace "atajo" ha dejado de existir si no se recibe un mensaje de un nodo reenviado a través de dicho atajo.

4.5.10.3 Tiempo de caducidad de las entradas en la tabla

Tanto en fallos en enlaces troncales (árbol), nodos o enlaces ajenos al árbol, los cambios topológicos no se realizan inmediatamente después de no recibir un mensaje, ya que puede deberse a un mensaje perdido o congestión en la red. Para ello se caracteriza cada entrada en la Tabla de Enlaces con un tiempo de caducidad (T_{cad}), desde que se añadió su marca de tiempo, pasado el cual se reacciona como se ha explicado anteriormente.

Se confirma la existencia de un fallo cuando este permanece durante un tiempo. Pasado este tiempo, se asume que, en el primer caso, el enlace troncal o el nodo se ha caído y hay que construir el árbol de nuevo o, en el segundo caso, que el enlace atajo ha dejado de existir y hay que modificar las tablas topológicas para no contemplarlo en el cálculo de rutas.

Para ello se implementa un temporizador para cada una de las entradas de la tabla. Este contador se reinicia cada vez que llega un mensaje de un nodo origen por el mismo puerto que se había recibido anteriormente.

Es importante establecer adecuadamente el periodo de vencimiento del temporizador, es decir, el tiempo de caducidad de las entradas de la tabla, T_{cad}. Éste deberá tener en cuenta un margen suficiente para asumir posibles mensajes perdidos o congestión en la red, pero sin ser demasiado amplio para poder reaccionar evitando inconsistencias en la red. Se puede demostrar que el valor mínimo satisfactorio de T_{cad} es de tres veces el tiempo de propagación máximo de un enlace en la red (T_r) [Rio12].

4.6 Comparación general de prestaciones

La comparación de prestaciones de los protocolos D-TRE entre sí (y con respecto al resto de protocolos de la familia) se ha realizado mediante un simulador desarrollado en Python. Basándonos en el mismo modelo de tráfico con de flujos unitarios desde cada uno de los N nodos que forman la red, hasta los N-1 nodos restantes, utilizado en los análisis de prestaciones individuales.

La simulación incluye tanto los protocolos D-TRE1 y D-TRE2, como también los protocolos Shortest Path, STP, TRE (extensión fija 1), TRE+ (extensión fija 2) y un hipotético 3TRE (extensión fija 3), para comparar los resultados obtenidos con los protocolos D-TRE.

El análisis de los resultados se realiza en base a las estadísticas extraídas del Simulador para cada clase de red (tipo de modelo de generador topológico, tamaño de la red y grado medio). Se realiza una comparativa de protocolos, mostrando gráficos por clase de red y protocolo.

4.6.1 Indicadores utilizados

Para caracterizar las prestaciones de los protocolos se analizan los cuatro parámetros que se han considerado como los indicadores más relevantes para este análisis y que permiten comparar los protocolos entre sí.

- Tamaño medio de las rutas: Es el factor determinante en la decisión de reenvío de los protocolos D-TRE. Estos protocolos se han diseñado con el requisito de ausencia de bucles. Una vez conseguido este objetivo, los nodos van a elegir un puerto concreto para reenviar cada trama en base a que el tamaño de la ruta hasta el destino sea el menor posible. Este indicador se mide en número de saltos, ya que se ha considerado que la distancia entre dos nodos unidos por un enlace es uno.

- Coste del encaminamiento: La decisión del encaminamiento, elección del puerto por el que reenviar una trama, lleva asociado un coste. Este coste es debido al cálculo de la distancia hasta el destino que proporciona cada nodo conocido y la comparación de estas distancias para obtener la menor. Este indicador se mide en número de consultas/comparaciones. En los protocolos que disponen de Tabla de Reenvío (ShP, TRE+, D-TRE1, D-TRE2 y un hipotético 3-TRE) se ha considerado el número de consultas que se realizan a dicha tabla. En los que no disponen de Tabla de Reenvío (STP y TRE) se ha estimado el número de comparaciones que son necesarias para reenviar una trama. En STP hay un único camino, por lo que será una sola consulta por nodo. En TRE se evalúan los vecinos, por lo que será el número de vecinos del nodo.

- Descarga de tráfico de los enlaces del árbol: Los protocolos de la familia TRE utilizan los enlaces no pertenecientes al árbol para encaminar tramas y obtener así un menor tamaño de la ruta. Este desvío del tráfico produce una liberación de carga en los enlaces del árbol, que son los más ocupados de la red. Este indicador se mide en porcentaje de flujos encaminados por los atajos, que proporciona el porcentaje de flujos que transcurren por enlaces no pertenecientes al árbol, respecto al total de flujos que circulan por la red.

- Rendimiento (Throughput) de la red, relativo a Shortest Path (ShP): El rendimiento de la red se evalúa con este indicador midiendo el tráfico que existe en el enlace más ocupado, que constituye el cuello de botella y es referencia para el diseño de una red. Este indicador se mide en porcentaje de flujos del enlace más ocupado de la red, respecto a Shortest Path,

que constituye el 100%. La referencia es Shortest Path porque es el protocolo que optimiza el tamaño de la ruta, pero no implica que sea el óptimo en desocupación del enlace más cargado.

4.6.2 Modelo de distribución de tráfico

El modelo de tráfico consiste en la generación de flujos unitarios para cada par origen-destino. Desde cada uno de los *N* nodos que forman la red, se genera un flujo hasta cada uno de los *(N-1)* nodos restantes. En total se generan *N(N-1)* flujos, que serán encaminados a través de la red por los nodos según el protocolo implementado. Los flujos son unitarios, es decir, de tasa unidad, sin carga real. La red, por lo tanto, es una red descargada, en la que no se produce ni congestión, ni bloqueo.

El simulador implementa para la topología en estudio, además del generador de *N(N-1)* flujos, el encaminamiento que rige a cada uno de los protocolos evaluados: Shortest Path, STP, TRE, TRE+, D-TRE1, D-TRE2 y 3-TRE

Por ejemplo, en una red de 512 nodos, se generan 261.632 flujos para cada uno de los siete protocolos, es decir, cada simulación genera casi 2 millones de flujos por topología. El número de flujos generados, según el número de nodos se muestra en la Tabla 51.

Tabla 51. Número de flujos generados según el número de nodos de la red

Nº nodos	Nº flujos generados por cada protocolo N(N-1)	Nº flujos generados por topología (7 protocolos)
64	4.032	28.224
128	16.256	113.792
256	65.280	456.960
512	261.632	1.831.424
1024	1.047.552	7.332.864

4.6.3 Redes simuladas

Se han modelado redes con topología aleatoria según las características de crecimiento sin escala (modelo Barabasi-Albert) y aleatoria pura (modelo Waxman), con grado medio de valores 4 y 6, y tamaños desde 64 hasta 1024 nodos. Un total de veinte combinaciones diferentes.

De cada una de ellas se han analizado entre cinco (redes grandes) y diez (redes pequeñas) topologías diferentes, eligiendo varias raíces del árbol y realizando una iteración para cada raíz. Las raíces elegidas corresponden con los *N* nodos de mayor grado. Por lo tanto, el umbral de elección de raíz es el enésimo nodo con mayor grado. En las redes pequeñas (con menor número de nodos) se han evaluado más raíces ($10 \leq N \leq 20$), que en las redes más grandes, en las que además, se han simulado menos topologías ($3 \leq N \leq 8$) ya que debido al número elevado de flujos generados (Tabla 51) el tiempo de simulación crecía enormemente.

Por ejemplo, de la clase [tipo=Barabasi, grado=4, nodos=128] se han simulado 10 topologías, realizándose en cada una de ella 20 iteraciones por cada raíz diferente. Por lo tanto, para esta clase de redes se han realizado 200 simulaciones, o dicho de otra manera, se han simulado 200 topologías.

4.6.4 Resultados comparados

A continuación se presenta una comparación de los indicadores del rendimiento de los protocolos simulados para las distintas clases de redes: el tamaño medio de las rutas seleccionadas, el coste de proceso del encaminamiento, el grado de utilización de enlaces ajenos al árbol (atajos) y el rendimiento relativo respecto a *Shortest Path*.

4.6.4.1 Tamaño medio de las rutas

Como se puede observar en las gráficas, los protocolos D-TRE introducen una mejora intermedia entre TRE+ (conocimiento de nodos situados a dos saltos de distancia) y el hipotético 3-TRE (a distancia tres saltos), siendo ligeramente mejor en D-TRE2 respecto a D-TRE1.

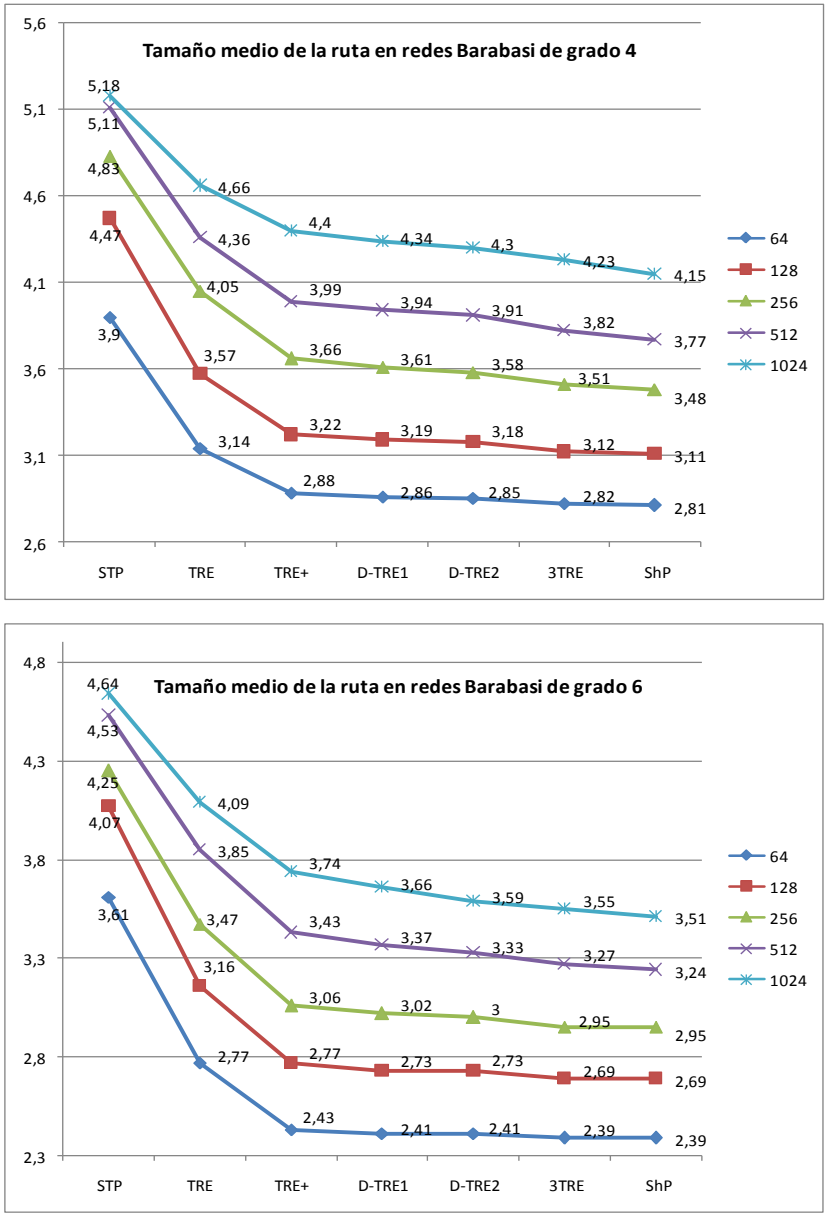


Ilustración 53. Tamaño medio de las rutas (nº de saltos) en redes Barabasi

Los protocolos D-TRE presentan valores muy similares a Shortest Path (ruta más corta, protocolo ideal para este parámetro) en redes de menor número de nodos, no siendo muy superior en redes de más nodos.

Por lo tanto, se pueden considerar los protocolos D-TRE como un punto intermedio entre los protocolos TRE+ y Shortest Path referido al tamaño medio de las rutas. Este valor se sitúa prácticamente en la media aritmética de ambos protocolos.

En todos los protocolos, cuanto mayor es el grado medio de la red, menor es el tamaño de la ruta, debido a que se incrementa el número de atajos, que reducen el número de saltos hasta el destino.

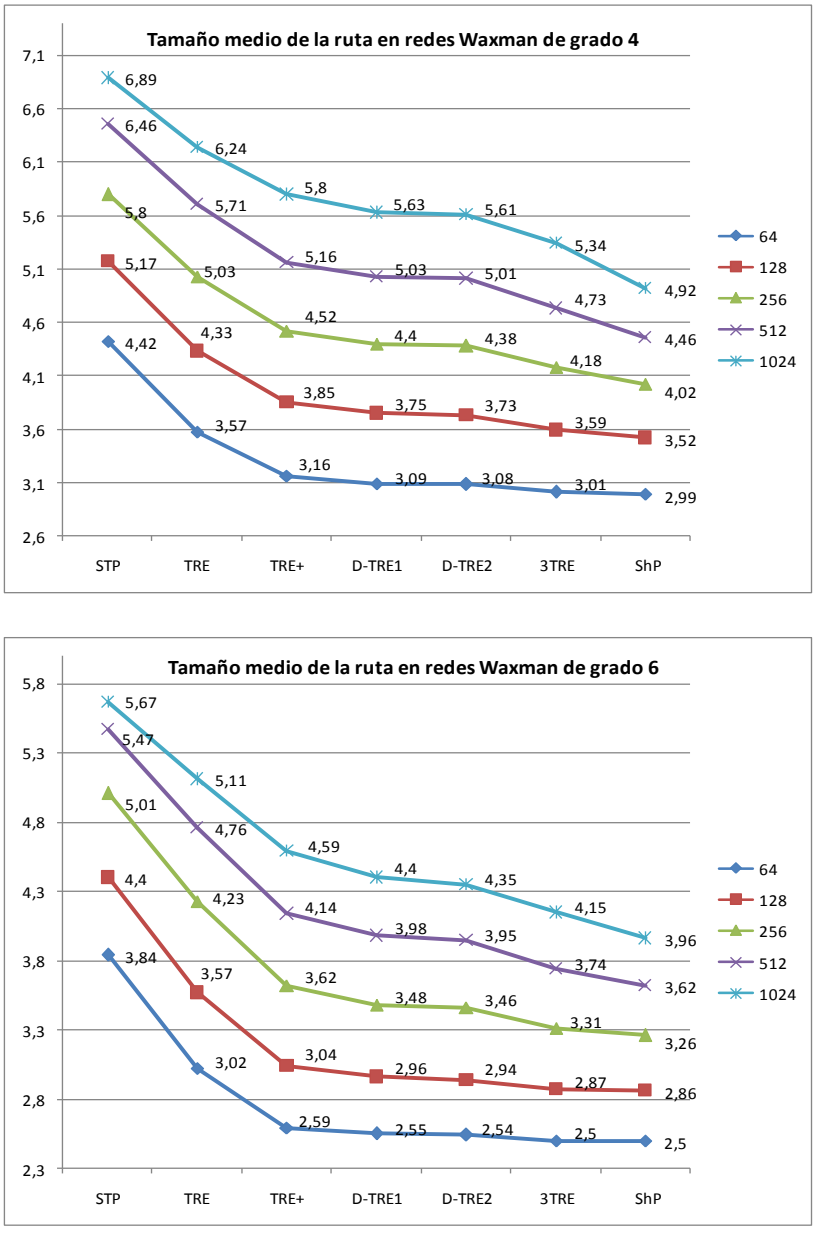


Ilustración 54. Tamaño medio de las rutas (nº de saltos) en redes Waxman

En las redes Barabasi-Albert (Ilustración 53) los caminos son más cortos para una misma topología en comparación con las redes Waxman (Ilustración 54), debido a que en la primera existe mayor concentración de enlaces por niveles (concretamente en los niveles superiores) que en las segundas y, por lo tanto, se utiliza mayor número de atajos que reducen el tamaño de la ruta.

La Tabla 52 y la Tabla 53 muestran los datos del tamaño medio de las rutas en las redes Barabasi y Waxman, respectivamente, que corresponden con los que aparecen en las gráficas de la Ilustración 54 y Ilustración 55.

Tabla 52. Tamaño medio de las rutas (nº de saltos) en redes Barabasi

Grado medio	Nº nodos	ShP	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	2,81	3,9	3,14	2,88	2,86	2,85	2,82
	128	3,11	4,47	3,57	3,22	3,19	3,18	3,12
	256	3,48	4,83	4,05	3,66	3,61	3,58	3,51
	512	3,77	5,11	4,36	3,99	3,94	3,91	3,82
	1024	4,15	5,18	4,66	4,4	4,34	4,30	4,23
6	64	2,39	3,61	2,77	2,43	2,41	2,41	2,39
	128	2,69	4,07	3,16	2,77	2,73	2,73	2,69
	256	2,95	4,25	3,47	3,06	3,02	3,00	2,95
	512	3,24	4,53	3,85	3,43	3,37	3,33	3,27
	1024	3,51	4,64	4,09	3,74	3,66	3,59	3,55

Tabla 53. Tamaño medio de las rutas (nº de saltos) en redes Waxman

Grado medio	Nº nodos	ShP	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	2,99	4,42	3,57	3,16	3,09	3,08	3,01
	128	3,52	5,17	4,33	3,85	3,75	3,73	3,59
	256	4,02	5,8	5,03	4,52	4,40	4,38	4,18
	512	4,46	6,46	5,71	5,16	5,03	5,01	4,73
	1024	4,92	6,89	6,24	5,8	5,63	5,61	5,34
6	64	2,5	3,84	3,02	2,59	2,55	2,54	2,5
	128	2,86	4,4	3,57	3,04	2,96	2,94	2,87
	256	3,26	5,01	4,23	3,62	3,48	3,46	3,31
	512	3,62	5,47	4,76	4,14	3,98	3,95	3,74
	1024	3,96	5,67	5,11	4,59	4,40	4,35	4,15

4.6.4.2 Coste del encaminamiento

Las gráficas representan el coste asociado a la decisión del puerto por el que se encamina una trama hasta un destino incluido en la ruta. Este coste se obtiene calculando el número de consultas que los nodos han realizado por ruta, desde el origen hasta el destino. Por ejemplo, en una ruta de cinco nodos, con diez consultas por nodo, el número de consultas será 50.

Se ha fijado escala logarítmica en las gráficas, ya que el número de consultas de *Shortest Path* es muy elevado, concretamente *(N-1)* nodos multiplicado por el número medio de saltos por ruta

(siendo N el número de nodos de la red), ya que en este protocolo los nodos deben conocer la red completa.

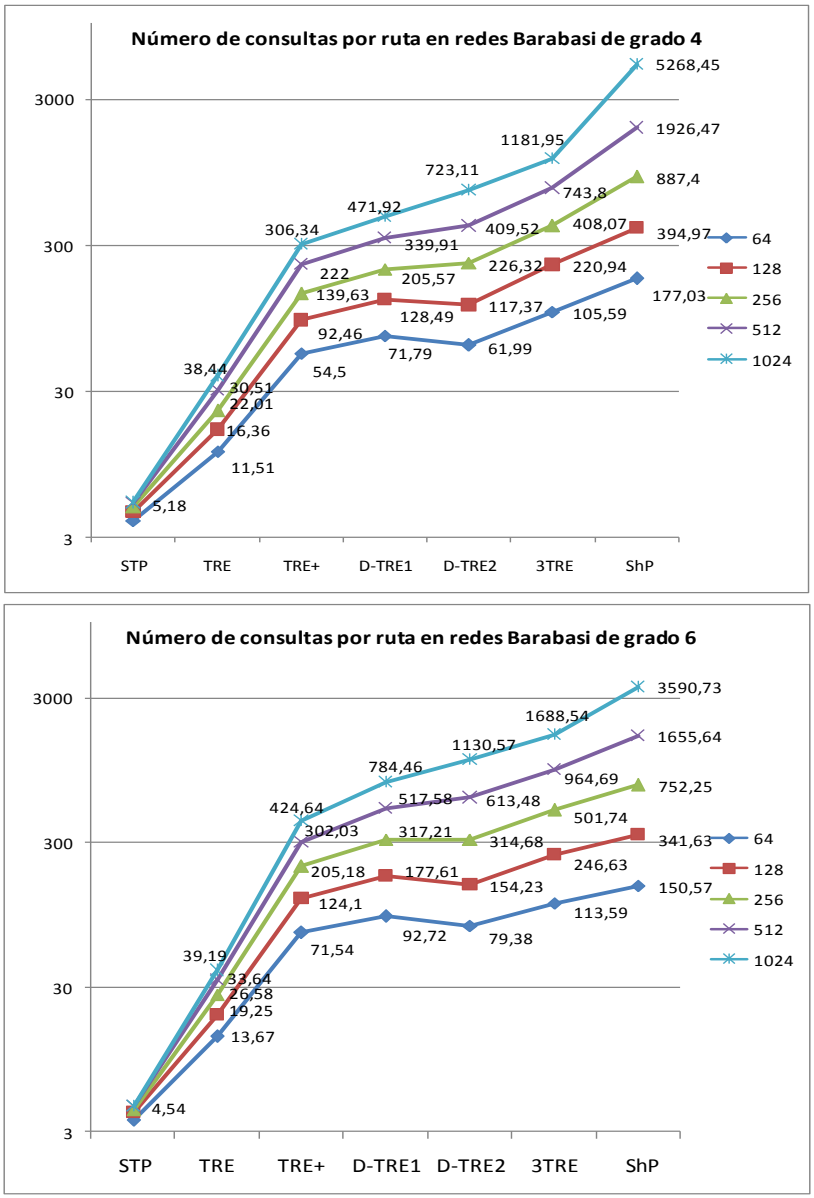


Ilustración 55. Coste del encaminamiento (nº de consultas/ruta) en redes Barabasi

Los protocolos D-TRE presentan valores ligeramente superiores a TRE+, sobre todo en redes pequeñas. Es significativa la mejora respecto a un hipotético 3-TRE y sobre todo respecto a *Shortest Path*, cuyo coste para conseguir la ruta más corta es muy elevado.

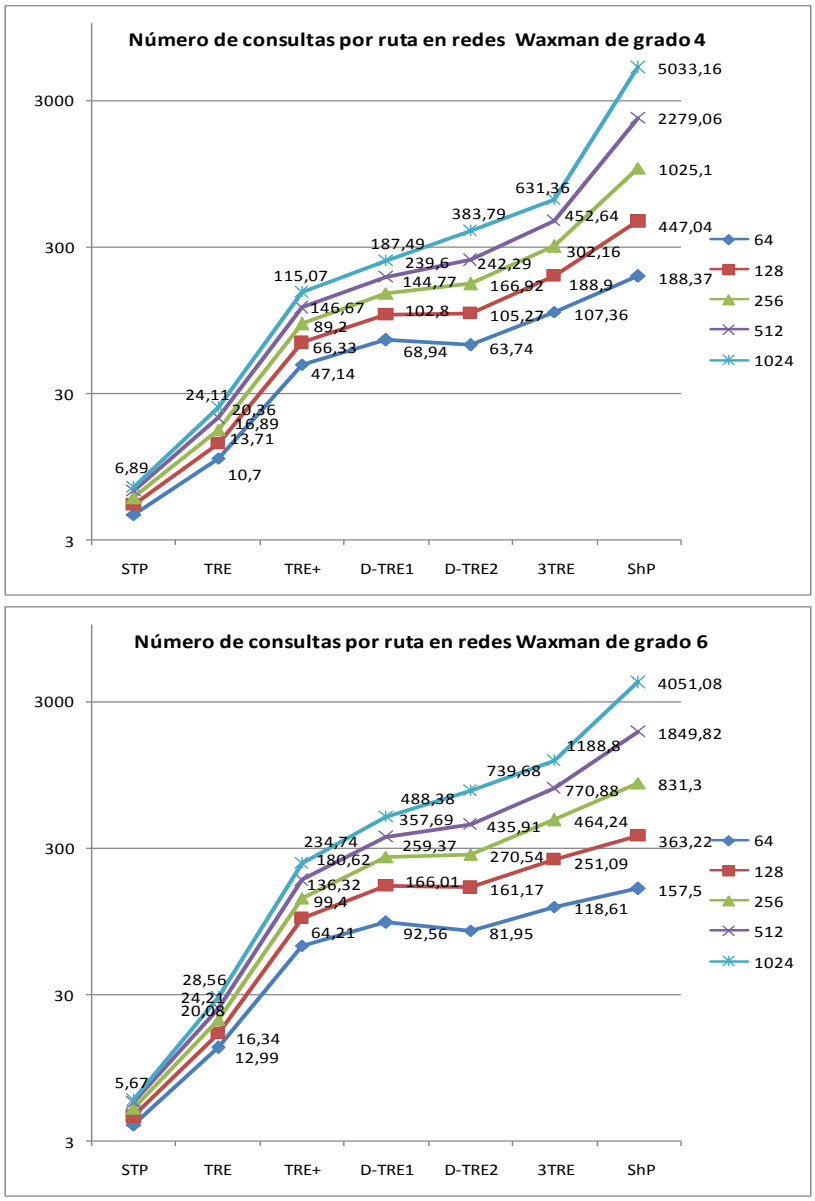


Ilustración 56. Coste del encaminamiento (nº de consultas/ruta) en redes Waxman

Comparando los protocolos D-TRE entre sí, se observa que D-TRE2 necesita menos consultas para encaminar la trama al destino que D-TRE1 en redes con menos de 256 nodos. Sin embargo, en redes con más de 256 nodos, D-TRE1 tiene menor coste que D-TRE2. Para redes de 256 nodos, el coste es prácticamente el mismo para ambos protocolos. Esto es debido a que cuanto mayor sea el número de nodos, más cantidad de ellos se sitúan en los niveles superiores, que es donde D-TRE2 posee el mayor alcance y, por lo tanto, el mayor número de atajos y las tablas más grandes. En cambio, en D-TRE1, el número de consultas depende del grado del nodo (y de sus vecinos) siendo independiente del nivel, por consiguiente, el coste crece proporcionalmente al número de nodos, pero no tan deprisa como en D-TRE2.

En todos los protocolos, excepto en *Shortest Path*, cuanto mayor es el grado medio de la red, mayor es el coste, debido a que se descubren más nodos y hay que comparar sus distancias. En *Shortest Path* no es así porque los nodos conocen igualmente toda la red, independientemente del grado.

Las Tabla 54 y Tabla 55 muestran los datos del tamaño medio de las rutas en las redes Barabasi y Waxman, respectivamente, que corresponden con los datos que aparecen en las gráficas de la Ilustración 57 y Ilustración 58.

Tabla 54. Coste del encaminamiento (nº de consultas/ruta) en redes Barabasi

Grado medio	Nº nodos	ShP	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	177,03	3,9	11,51	54,5	71,79	61,99	105,59
	128	394,97	4,47	16,36	92,46	128,49	117,37	220,94
	256	887,4	4,83	22,01	139,63	205,57	226,32	408,07
	512	1926,5	5,11	30,51	222	339,91	409,52	743,8
	1024	5268,5	5,18	38,44	306,34	471,92	723,11	1182
6	64	150,57	3,61	13,67	71,54	92,72	79,38	113,59
	128	341,63	4,07	19,25	124,1	177,61	154,23	246,63
	256	752,25	4,25	26,58	205,18	317,21	314,68	501,74
	512	1655,6	4,53	33,64	302,03	517,58	613,48	964,69
	1024	3590,7	4,54	39,19	424,64	784,46	1130,6	1688,5

Tabla 55. Coste del encaminamiento (nº de consultas/ruta) en redes Waxman

Grado medio	Nº nodos	ShP	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	188,37	4,42	10,7	47,14	68,94	63,74	107,36
	128	447,04	5,17	13,71	66,33	102,8	105,27	188,9
	256	1025,1	5,8	16,89	89,2	144,77	166,92	302,16
	512	2279,1	6,46	20,36	115,07	187,49	242,29	452,64
	1024	5033,2	6,89	24,11	146,67	239,6	383,79	631,36
6	64	157,5	3,84	12,99	64,21	92,56	81,95	118,61
	128	363,22	4,4	16,34	99,4	166,01	161,17	251,09
	256	831,3	5,01	20,08	136,32	259,37	270,54	464,24
	512	1849,8	5,47	24,21	180,62	357,69	435,91	770,88
	1024	4051,1	5,67	28,56	234,74	488,38	739,68	1188,8

4.6.4.3 Descarga de tráfico por enlaces atajo

Las siguientes cuatro gráficas muestran el porcentaje de flujos que son encaminados a través de los atajos, es decir, el porcentaje de flujos que los protocolos de la familia TRE liberan del árbol, descargando sus enlaces troncales. La referencia (el 100%) es el número total de flujos por enlace existentes en la red. El protocolo *Shortest Path* no se incluye en las gráficas porque no es un protocolo basado en árbol de expansión. El protocolo STP no utiliza atajos, por lo que este porcentaje es cero.

En la Ilustración 57 se refleja el porcentaje de flujos que son encaminados por atajos, en lugar de ocupar los enlaces del árbol, en las Redes Barabasi de grado medio 4 y 6.

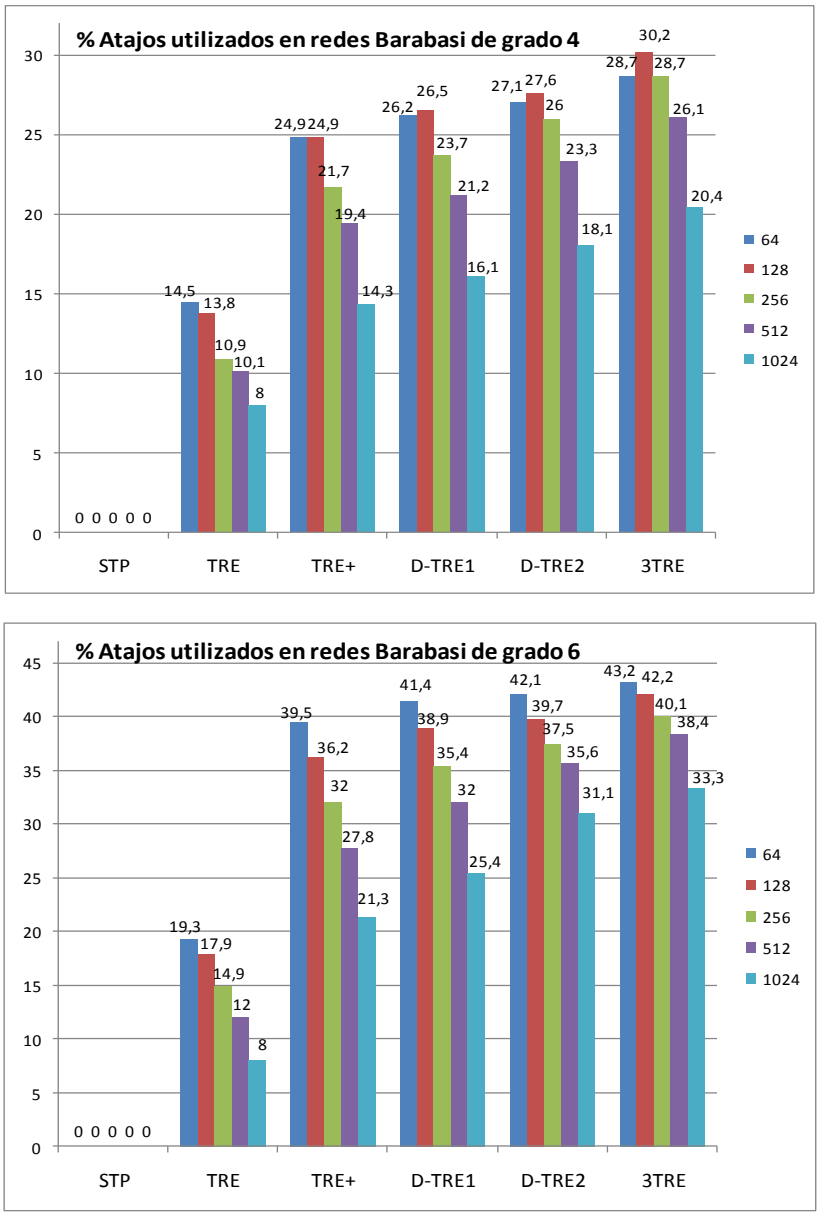


Ilustración 57. Descarga de tráfico del árbol (Redes Barabasi)

En la Ilustración 58 se refleja el porcentaje de flujos que son encaminados por atajos, en redes Waxman de grado medio 4 y 6.

Se comprueba que en las redes de menor grado medio, las redes Barabasi desvían más porcentaje de flujos a través de los atajos que las redes Waxman. Sin embargo, las segundas obtienen mejor resultado que las primeras en redes de mayor grado medio. Anteriormente habíamos observado que las redes Barabasi consiguen rutas más cortas que las Waxman en todo tipo de redes.

La explicación de que las redes Barabasi desvíen menos porcentaje de tráfico fuera del árbol que las Waxman en redes con mayor número de vecinos por nodo, pero que aun así consigan rutas más cortas, radica en que los enlaces que toman son más efectivos. Si los *'b'* atajos Barabasi se acercan más al destino que los '*w*' atajos Waxman, considerando $b > w$, la ruta Barabasi tendrá menor tamaño, mientras que la utilización de los atajos será mayor en Waxman. Por ejemplo, pongamos una red Waxman en la que se encamina la trama hasta llegar al destino a través de seis enlaces del árbol y tres atajos, y una red Barabasi en la que se encamina la trama por siete enlaces del árbol y un solo

atajo. En la red Waxman el tamaño de la ruta es de 9 saltos y el porcentaje de tráfico desviado del árbol es de un 33%. En cambio, en la red Barabasi, la ruta es menor, 8 saltos, pero el porcentaje de tráfico descargado del árbol es de un 13%.

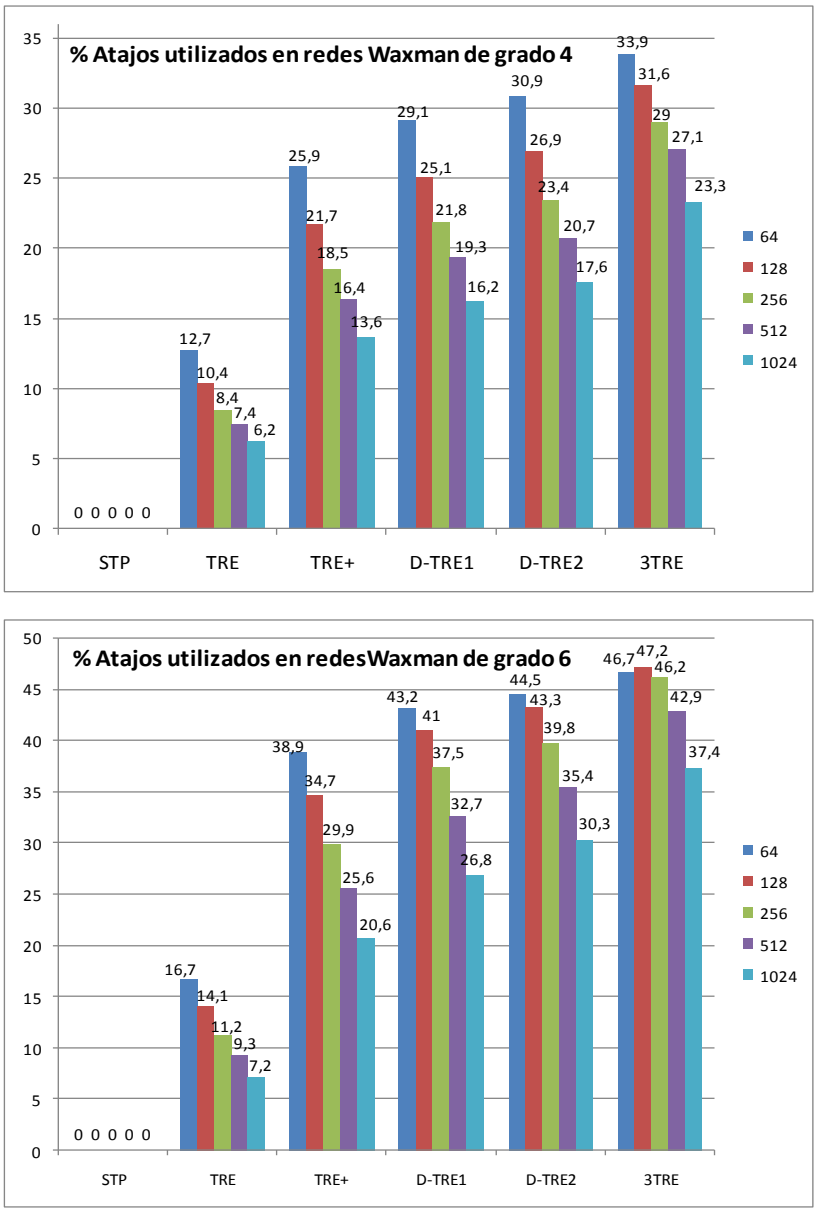


Ilustración 58. Descarga de tráfico del árbol (Redes Waxman)

Esto es debido a que en las redes Barabasi, al aumentar el grado, algunos nodos actúan como *hubs* que conectan con gran cantidad de nodos, por lo que aumenta la probabilidad de un mayor acercamiento al nodo destino o su rama. Por otro lado, en las redes Waxman se distribuyen los vecinos más uniformemente, pudiendo encadenar varios nodos de grado alto (pero no tan altos como el *hub* de la red Barabasi).

Se observa en las gráficas que D-TRE introduce una ganancia respecto a TRE+ (dos saltos de alcance) similar a la que consigue el hipotético 3-TRE (tres saltos de alcance) respecto a D-TRE. Esta linealidad en la ganancia se comprueba en las Tabla 56 y Tabla 57, que muestran la ganancia del porcentaje de carga liberada del árbol respecto al protocolo TRE+ para las distintas clases de redes estudiadas.

Tabla 56. Ganancia en el *"uso de atajos"* respecto a TRE+ (redes con grado medio 4)

	Redes Barabasi					Redes Waxman				
	64	128	256	512	1024	64	128	256	512	1024
TRE+	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
D-TRE1	1,05	1,06	1,09	1,09	1,13	1,12	1,16	1,18	1,18	1,19
D-TRE2	1,09	1,11	1,20	1,20	1,27	1,19	1,24	1,26	1,26	1,29
3TRE	1,15	1,21	1,32	1,35	1,43	1,31	1,46	1,57	1,65	1,71

Tabla 57. Ganancia en el *"uso de atajos"* respecto a TRE+ (redes con grado medio 6)

	Redes Barabasi					Redes Waxman				
	64	128	256	512	1024	64	128	256	512	1024
TRE+	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
D-TRE1	1,05	1,07	1,11	1,15	1,19	1,11	1,18	1,25	1,28	1,30
D-TRE2	1,07	1,10	1,17	1,28	1,46	1,14	1,25	1,33	1,38	1,47
3TRE	1,09	1,17	1,25	1,38	1,56	1,20	1,36	1,55	1,68	1,82

En la comparación entre protocolos, D-TRE2 utiliza más porcentaje de atajos que D-TRE1. Este mejor aprovechamiento de la red coincide con un mejor rendimiento de la red en referencia al tamaño de las rutas D-TRE2 respecto a D-TRE1 (apartado 8.2.1). La mejora del porcentaje de atajos usados en D-TRE2 se hace evidente a medida que aumenta el número de nodos de la red y su grado medio, debido a que se descubre más cantidad de nodos en todos los niveles del árbol, sobre todo en los superiores que es donde los nodos D-TRE2 tienen mayor alcance.

Las Tabla 58 y Tabla 59 muestran los datos del porcentaje de atajos utilizados en las redes Barabasi y Waxman, respectivamente, que son los que aparecen en las gráficas de la Ilustración 57 y Ilustración 58.

Tabla 58. % de flujos encaminados por atajos (Redes Barabasi)

Grado medio	Nº nodos	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	0	14,5	24,9	26,2	27,1	28,7
	128	0	13,8	24,9	26,5	27,6	30,2
	256	0	10,9	21,7	23,7	26	28,7
	512	0	10,1	19,4	21,2	23,3	26,1
	1024	0	8	14,3	16,1	18,1	20,4
6	64	0	19,3	39,5	41,4	42,1	43,2
	128	0	17,9	36,2	38,9	39,7	42,2
	256	0	14,9	32	35,4	37,5	40,1
	512	0	12	27,8	32	35,6	38,4
	1024	0	8	21,3	25,4	31,1	33,3

Tabla 59. % de flujos encaminados por atajos (Redes Waxman)

Grado medio	Nº nodos	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	0	12,7	25,9	29,1	30,9	33,9
	128	0	10,4	21,7	25,1	26,9	31,6
	256	0	8,4	18,5	21,8	23,4	29
	512	0	7,4	16,4	19,3	20,7	27,1
	1024	0	6,2	13,6	16,2	17,6	23,3
6	64	0	16,7	38,9	43,2	44,5	46,7
	128	0	14,1	34,7	41	43,3	47,2
	256	0	11,2	29,9	37,5	39,8	46,2
	512	0	9,3	25,6	32,7	35,4	42,9
	1024	0	7,2	20,6	26,8	30,3	37,4

4.6.4.4 Rendimiento (*Throughput*) relativo a *Shortest Path*

Directamente relacionada con el aumento del porcentaje de atajos utilizados para encaminar tramas está la descongestión del enlace más ocupado de la red, que suele estar en torno a la raíz del árbol. Este es el indicador de rendimiento mostrado en las siguientes gráficas.

Se observa en la Ilustración 59 cómo, en redes Barabasi, el *Throughput* de los protocolos D-TRE, es muy cercano al 100% en redes con menor número de nodos (es incluso mayor de 100 en redes de 64 nodos y grado medio 6). A medida que aumenta el número de nodos, la nueva distribución de la red, más expandida, obliga a un mayor número de flujos a ascender hasta la raíz para poder alcanzar su destino, transcurriendo por el enlace más ocupado, y disminuyendo el *Throughput*.

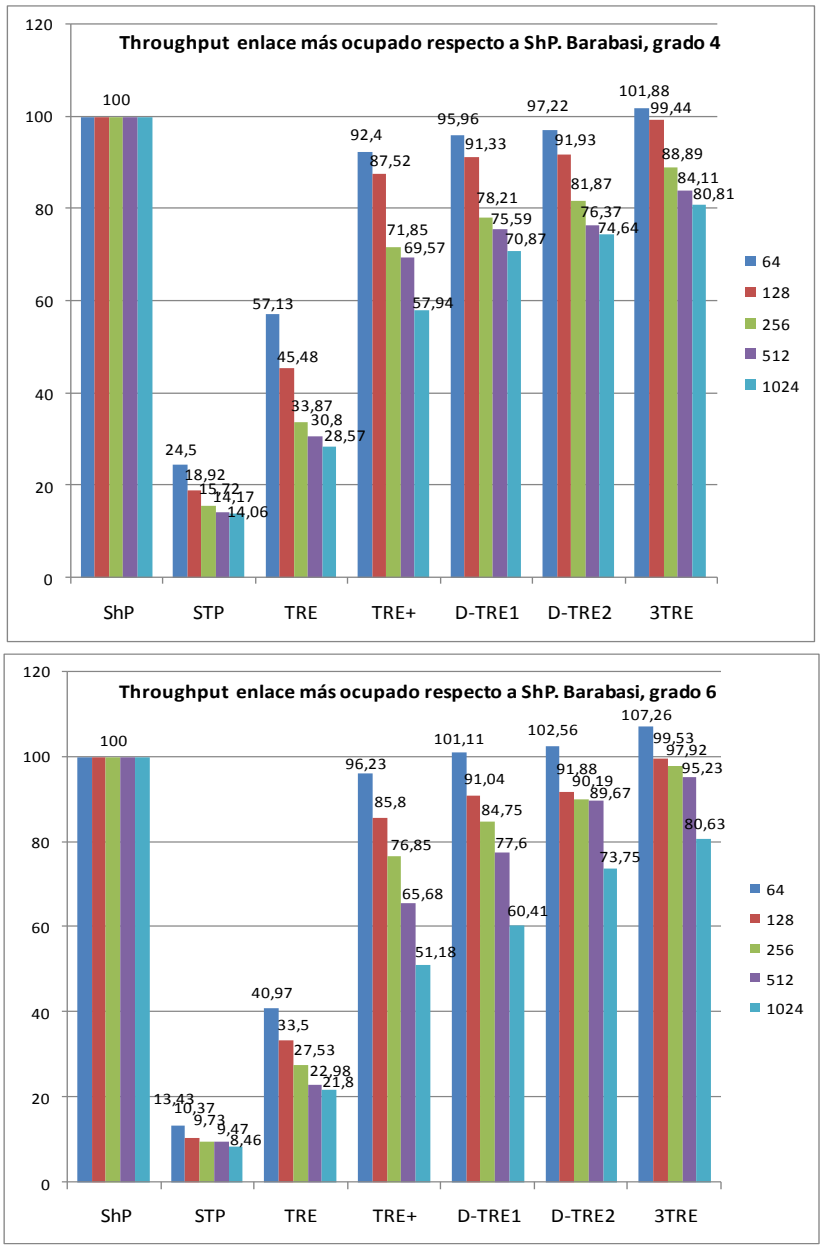


Ilustración 59. Throughput del número de flujos en el enlace más ocupado, respecto a ShP (Redes Barabasi)

Se observa en la Ilustración 60 cómo se obtienen, en general, peores resultados para redes Waxman respecto a redes Barabasi. Esto es debido a que se elige como raíz del árbol al nodo de mayor grado. En Barabasi, este nodo tiene un número de vecinos considerablemente mayor que en Waxman. Por lo tanto, los nodos que tienen que atravesar la raíz irremediablemente disponen de más enlaces por los que ser encaminados, mientras que en Waxman, al haber menos enlaces descendentes de la raíz, tienen que pasar obligatoriamente por el enlace más ocupado.

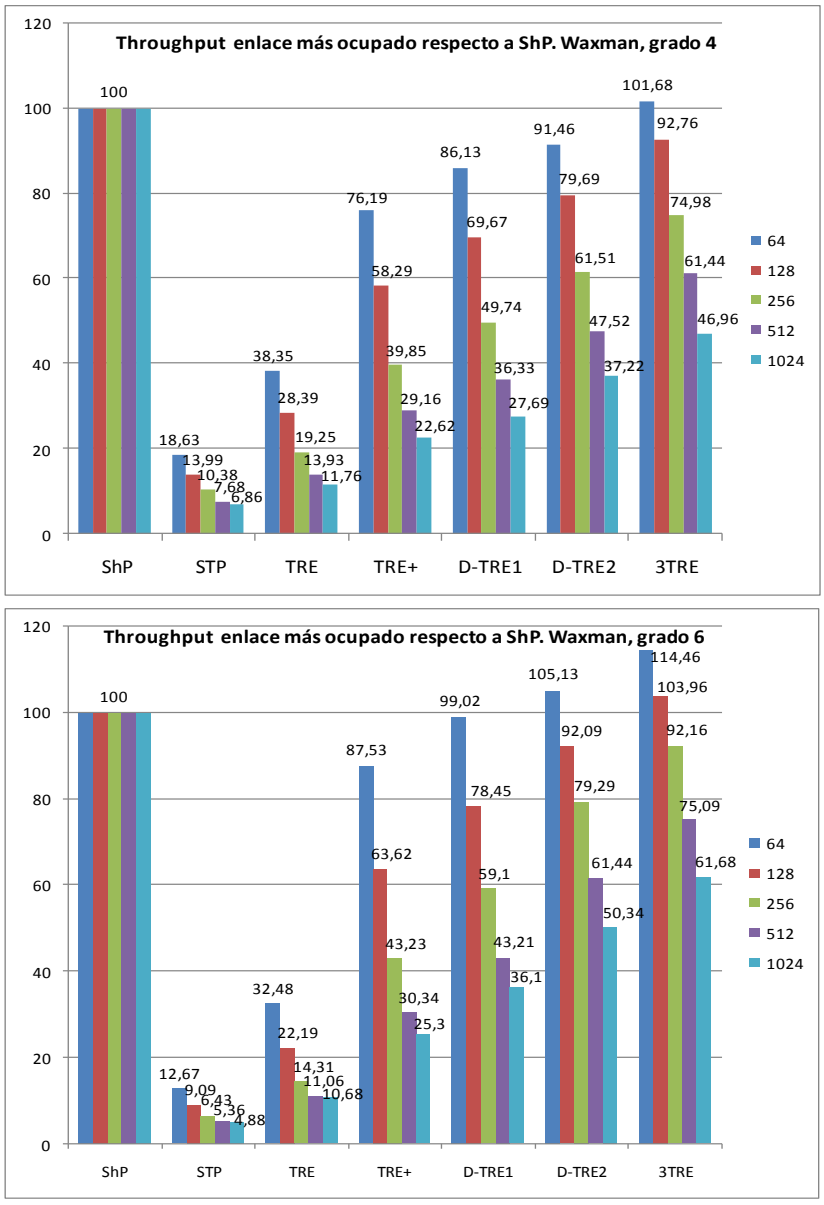


Ilustración 60. Throughput del número de flujos en el enlace más ocupado, respecto a ShP (Redes Waxman)

Los resultados obtenidos en redes de grado medio *d=6* son considerablemente superiores respecto a redes de grado medio *d=4*, debido a que hay más atajos y, por lo tanto, más probabilidad de evitar el enlace más ocupado.

Las tablas Tabla 60 y Tabla 61 muestran los datos de Throughput en redes Barabasi y Waxman, respectivamente, son los contenidos en las gráficas de la Ilustración 59 y Ilustración 60.

Tabla 60. Rendimiento relativo a *Shortest Path* (ShP) en Redes Barabasi

Grado medio	Nº nodos	ShP	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	100	24,5	57,13	92,4	95,96	97,22	101,88
	128	100	18,92	45,48	87,52	91,33	91,93	99,44
	256	100	15,72	33,87	71,85	78,21	81,87	88,89
	512	100	14,17	30,8	69,57	75,59	76,37	84,11
	1024	100	14,06	28,57	57,94	70,87	74,64	80,81
6	64	100	13,43	40,97	96,23	101,11	102,56	107,26
	128	100	10,37	33,5	85,8	91,04	91,88	99,53
	256	100	9,73	27,53	76,85	84,75	90,19	97,92
	512	100	9,47	22,98	65,68	77,6	89,67	95,23
	1024	100	8,46	21,8	51,18	60,41	73,75	80,63

Tabla 61. Rendimiento relativo a *Shortest Path* (ShP) en redes Waxman

Grado medio	Nº nodos	ShP	STP	TRE	TRE+	D-TRE1	D-TRE2	3TRE
4	64	100	18,63	38,35	76,19	86,13	91,46	101,68
	128	100	13,99	28,39	58,29	69,67	79,69	92,76
	256	100	10,38	19,25	39,85	49,74	61,51	74,98
	512	100	7,68	13,93	29,16	36,33	47,52	61,44
	1024	100	6,86	11,76	22,62	27,69	37,22	46,96
6	64	100	12,67	32,48	87,53	99,02	105,13	114,46
	128	100	9,09	22,19	63,62	78,45	92,09	103,96
	256	100	6,43	14,31	43,23	59,1	79,29	92,16
	512	100	5,36	11,06	30,34	43,21	61,44	75,09
	1024	100	4,88	10,68	25,3	36,1	50,34	61,68

4.7 Conclusiones

Comparando los protocolos, tanto D-TRE1 como D-TRE2 presentan una mejora respecto a TRE+, con mayor cercanía al 100% (objetivo teórico ofrecido por *Shortest Path*) cuanto menor es el número de nodos y mayor es el grado medio de la red. Es destacable que en redes de 64 nodos y grado medio *d=6*, los protocolos D-TRE se comportan más eficientemente que *Shortest Path*. El protocolo D-TRE2 obtiene mejores resultados que D-TRE1 debido a la existencia de nodos de mayor grado en los niveles superiores, consiguiendo evitar el enlace más ocupado. Todos ellos presentan baja sobrecarga.

5 El protocolo ARP-Path. Simulación de flujos

Este capítulo describe la tercera y cuarta contribuciones principales de esta Tesis. Se enmarcan dentro de la línea de investigación principal del grupo GIST-Netserv, línea centrada desde el año 2009 en el desarrollo de una nueva familia de puentes transparentes de camino mínimo denominados genéricamente All-Path porque encuentran el camino más corto por exploración simultánea de todos los caminos. Se presenta el primer protocolo de la familia, denominado Fast-Path (posteriormente rebautizado como ARP-Path), al que el autor ha contribuido desde sus inicios en los que se exploraba la (hasta entonces considerada imposible) factibilidad de la inundación total de las tramas de difusión en una red sin bloquear enlaces y sin que se produjeran bucles. Se detalla el mecanismo de control de bucles diseñado para sustituir al árbol de expansión (cuya patente acaba de ser reconocida al grupo y se encuentra en proceso de internacionalización) y se presenta el entorno de caracterización del rendimiento junto con los excelentes resultados obtenidos en su evaluación. Un elemento clave desarrollado en esta contribución es el simulador de flujos de ARP-Path que facilita caracterizar su comportamiento con el suficiente nivel de abstracción y requisitos temporales y computacionales asequibles que está permitiendo mostrar la sorprendente capacidad nativa de reparto de carga del protocolo ARP-Path, derivada de establecer los caminos justo cuando se precisan y en función de la latencia instantánea de la red.

5.1 Introducción

Las redes metropolitanas o redes campus basadas en Ethernet ofrecen ventajas tales como sencillez de configuración (incluso autoconfiguración completa), la de evitar la administración de direcciones IP en subredes, alto rendimiento y bajo coste. Actualmente se basan en los estándares establecidos de protocolos de árbol de expansión como el protocolo *Rapid Spanning Tree* (RSTP) y *Multiple Spanning Tree* (MSTP) o en la utilización de topologías libres de bucles. Sin embargo, este tipo de redes no son escalables por las grandes limitaciones impuestas por el protocolo de árbol de expansión utilizado para evitar bucles y por la necesidad de segmentar la red en distintas subredes IP mediante enrutadores para limitar la zona de fallo ante una tormenta de tramas de difusión en capa dos. Para mejorar la escalabilidad de las redes campus Ethernet, hasta formar una subred IP única de forma fiable, existen propuestas en estandarización tales como *Routing Bridges* [Per04] en el grupo TRILL del IETF, y Shortest Path Bridges [Fin05] en el grupo IEEE 802.1aq.

ARP-Path es una evolución conceptual de los puentes transparentes con aprendizaje que no requiere protocolo auxiliar de encaminamiento (ni de árbol de expansión) en capa dos, a diferencia de las propuestas actualmente en estandarización RBridges y Shortest Path Bridges. Cada host establece, en los conmutadores ARP-Path, un camino mínimo al mismo tiempo que se envía el paquete estándar *ARP-Request*, pero inundado por todos los enlaces. El camino marcado por el paquete *ARP-Request* que alcanza el destino se confirma aprovechando el paquete *ARP-Reply* de respuesta del host destino.

Se han realizado implementaciones de puentes ARP-Path en Linux y en la plataforma *Openflow* [OFS] con tarjetas *NetFPGA* [NetFPGA] (actualmente en pruebas), así como simulaciones sobre Omnet++[OMNET]. Las prestaciones son similares o ligeramente superiores a las de enrutadores de camino mínimo y muy superiores a RSTP. Las pruebas con tráfico real y de reconfiguración muestran las robustez y rapidez del protocolo. ARP-Path no modifica la trama de Ethernet y es compatible con puentes estándar en modo núcleo-isla.

5.1.1 Antecedentes

Se puede considerar que el antecedente conceptual del protocolo ARP-Path es *Reverse Path Forwarding* [DM78], un mecanismo para la difusión multicast en árbol que utiliza las tablas de encaminamiento unicast de los enrutadores para validar el origen del paquete multicast recibido. El principio de funcionamiento se basa en que el enrutador reenvía por todos los demás puertos los paquetes recibidos cuya dirección IP unicast origen esté asociada en la tabla de rutas del enrutador como destino por la interfaz por la que llegó. El enrutador descartará los recibidos por otras interfaces. Asimismo, los protocolos que descubren rutas mediante inundación como AODV [RFC3561] presentan alguna característica en común con ARP-Path.

5.2 El protocolo ARP-Path

ARP-Path aprovecha la propia difusión de las tramas de control del protocolo ARP (en concreto la trama *ARP-Request)* para encontrar 'sobre la marcha' el camino de menor latencia que existe en ese momento en la red entre dos sistemas origen y destino que quieren intercambiar tráfico. El mecanismo de difusión controlado que se ha diseñado permite explorar todos los caminos posibles y

detectar de entre ellos el de menor latencia (por tanto, es un enrutamiento de tipo camino más corto, *shortest path*), garantizando a la vez que no se producen bucles en el proceso. Podríamos decir que ARP-Path es un protocolo que opera en el plano de datos, al no requerir un protocolo de control específico (salvo para acelerar la reconstrucción de caminos tras fallos en la red).

Hay tres diferencias básicas entre los puentes ARP-Path y los puentes transparentes estándar de árbol de expansión: en primer lugar, ARP-Path no bloquea enlaces limitando el número de los enlaces utilizados a una topología de árbol; en segundo lugar, modifica el mecanismo de aprendizaje de direcciones en los puertos, introduciendo el importante concepto del 'bloqueo del aprendizaje (*lock*)' de la dirección aprendida en el primer puerto que recibe la trama; y tercero, evita la replicación por todos los puertos de las tramas con dirección de destino desconocida mediante un mecanismo que reconstruye el camino dañado.

5.2.1 Descubrimiento de camino (*ARP-Request*)

El mecanismo de creación de rutas de acceso rápido está parcialmente inspirado en el *Reverse Path Forwarding*. Un camino ARP-Path es el camino más rápido (y por tanto, único) creado por la primera copia de una trama de petición ARP que alcanza su destino. El procedimiento, descrito en la Ilustración 61, funciona como se describe a continuación.

El host S envía una petición ARP encapsulada en una trama de difusión B para resolver la dirección IP de un host de destino D. El conmutador frontera 2 recibe la trama de S y asocia la dirección MAC de S al puerto a través del cual se ha recibido la trama, ligando temporalmente el aprendizaje de la dirección S a este puerto y bloqueando su aprendizaje por los demás puertos del puente 2 al recibir tramas con origen S. Por lo tanto, las tramas con dirección de origen S, que lleguen a otros puertos del conmutador 2, son descartadas por llegar después de la primera. Después, el conmutador 2 reenvía la trama por todos los puertos excepto por el que se recibió.

Los conmutadores 3 y 1 hacen lo mismo que el conmutador 2, asociando la dirección S al puerto por el que primero reciben la trama. Luego, también asocian la dirección S al puerto por donde se recibe primero y reenvían la trama por todos los puertos excepto el puerto por el que llegó, por lo que las copias duplicadas de la trama B llegan a 3 y 1 respectivamente, reenviadas por 1 y 3. Sin embargo, estas tramas llegan a un puerto diferente del puerto asociado y bloqueado temporalmente a S, por lo que se descartan por su dirección origen S, sin necesidad de detección de tramas duplicadas mediante otros mecanismos.

Lo mismo ocurre en los conmutadores 4 y 5. Por lo tanto, la asociación temporal (bloqueo) de la dirección de S a un puerto en cada conmutador se propaga a través de la red como un árbol enraizado en S, hasta que se alcanzan los conmutadores frontera de la red y los hosts conectados a ellos, incluyendo el host D, destino del *ARP-Request*. Queda así seleccionada una cadena de conmutadores entre S y D que tiene un puerto de entrada asociado a S con el aprendizaje de esa dirección bloqueado para otros puertos durante un tiempo.

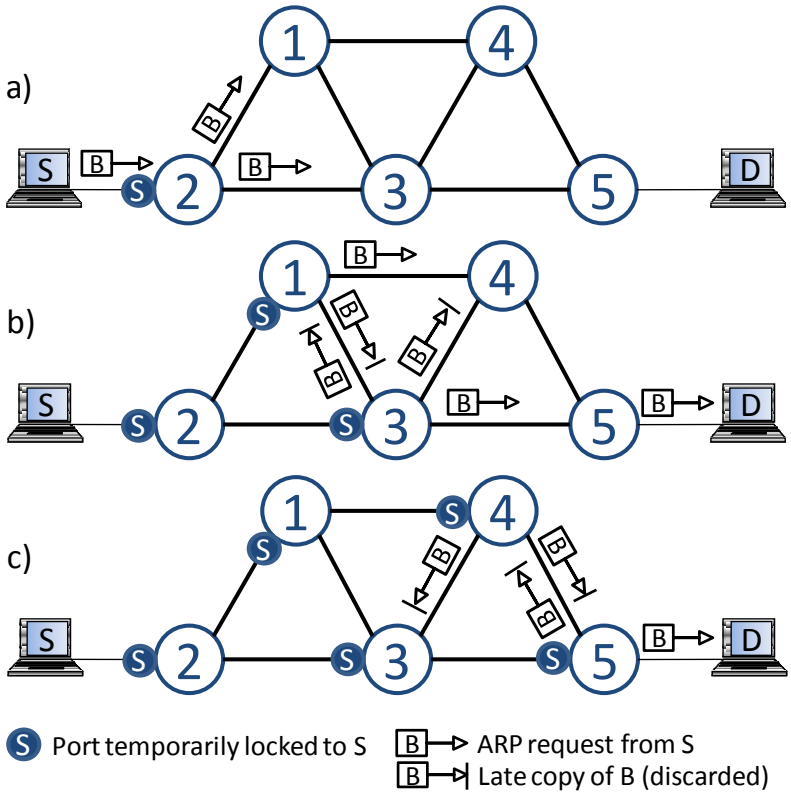


Ilustración 61. ARP-Path, fase de descubrimiento de camino entre S y D

Como hemos dicho antes, un camino ARP-Path es el camino más rápido (y, por tanto, único), creado por la primera copia (y única, como se verá más adelante) de la petición ARP que llega al destino solicitado, por carrera entre todos los enlaces de la red. Básicamente, cuando el host S (fuente) envía un ARP Request en difusión para resolver la dirección IP de un host D (destino), éste se difunde por toda la topología y cada puente lo recibe por uno o más puertos.

Así pues, el camino "más rápido" se genera en base a que los conmutadores capturan (y reenvían en modo difusivo) la primera copia del *ARP-Request* que reciben, es decir, la que llegó primero, asocian el puerto por el que llegó a la dirección MAC origen de la trama y le asignan el estado *locked* al par <dirección IP-puerto>, bloqueando el resto de puertos para las tramas que se reciban del mismo origen, por indicar caminos "más lentos" y ser tramas duplicadas, redundantes en el árbol de inundación del *ARP-Request* formado desde el host origen hasta el resto de los hosts.

Esto sucede en todos los conmutadores hasta que el *ARP-Request* alcanza el host destino. De manera que los conmutadores entre S y D han quedado con uno de sus puertos en estado bloqueado para esa dirección (*locked*). Este estado posee un temporizador que establece un tiempo durante el que se podrá pasar el puerto a estado 'confirmado' (*confirmed*), o se liberará si vence la temporización de bloqueo sin haber sido confirmado el camino, esto último sucederá para todas las ramas del árbol de inundación creadas por el *ARP-Request* excepto la rama que se confirme con la respuesta del host destino, como se indica a continuación.

5.2.2 Confirmación de camino (*ARP-Reply*)

El mecanismo de selección de camino en dirección contraria (confirmación del camino de ida) se describe en la Ilustración 62. El host D responde con un *ARP-Reply* unicast hacia el host S. El

conmutador 5 asocia la dirección de D al puerto de entrada y bloquea el aprendizaje de la dirección D en los demás puertos. Luego reenvía la trama por el puerto asociado a S. El conmutador 5 tiene ahora rutas confirmadas a S y a D. Los temporizadores caché se activan para las direcciones S y D en la tabla. El conmutador 3 asocia la dirección D al puerto de entrada y bloquea el aprendizaje de D en los demás puertos. Luego reenvía la trama por el puerto asociado a la dirección S, confirmando el camino y activando los temporizadores de caché para S y D. El conmutador 2 asocia la dirección D al puerto de entrada y bloquea el aprendizaje de la dirección D en los demás puertos, luego reenvía la trama a través del puerto asociado a S hasta el host S, confirmando el camino y activando los temporizadores de caché para S y D.

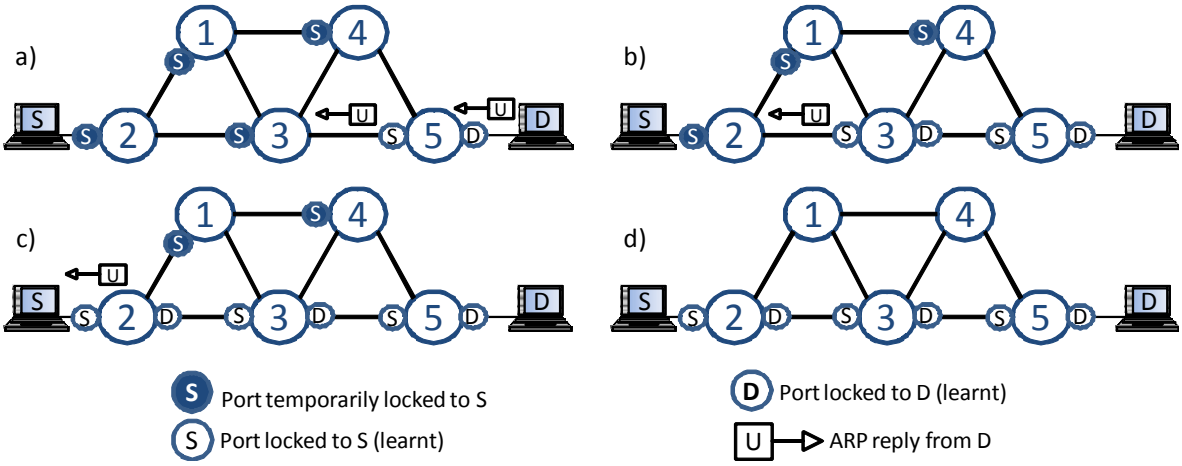


Ilustración 62. ARP-Path, fase de confirmación del camino entre S y D

Este mecanismo asegura que el camino es simétrico. El camino en la dirección S->D se confirma hacia atrás mediante la respuesta unicast del destino sobre el mismo en dirección D->S, es decir, algunos de los puertos *locked* anteriores pasarán a un estado confirmado (*confirmed)* y otros caducarán (cuando expire su temporizador), mientras que a su vez el host D asociará nuevos puertos a su MAC, directamente confirmados, mediante la respuesta ARP. Finalmente, los puertos c*onfirmed* tendrán a su vez un temporizador, que se irá refrescando con las nuevas tramas que por ellos pasen con direcciones origen y destino S y D, respectivamente.

5.2.3 Borrado de direcciones MAC por reconfiguración

En los conmutadores estándar, cuando falla un enlace las direcciones MAC aprendidas asociadas al puerto correspondiente se borran. ARP-Path realiza algo parecido pero con menor impacto en la red. El fallo de un conmutador puede ser detectado por la capa física en los puertos de entrada de sus nodos vecinos como un fallo de enlace. Se pueden utilizar mensajes estándar de comprobación de continuidad para detectar fallos de nodo o de enlace, como se define en las especificaciones ITU-T Y.1731 [Y1731] y e IEEE 802.1ag [8021ag]. Cuando el fallo de un enlace se detecta en cada uno de los puentes situados en los extremos del enlace, se activa en cada conmutador el borrado de todas las direcciones MAC asociadas con dicho puerto. Lo mismo sucede, en todos los puertos de un nodo, cuando éste se reinicializa. Los caminos aprendidos que persistían activados por los temporizadores de aprendizaje ya no son válidos y serán reparados por el procedimiento de reparación de caminos, pero sólo cuando sea necesario cada uno de ellos (no se repararán caminos que no vuelvan a necesitarse).

La dinámica de las direcciones MAC aprendidas en los conmutadores ARP-Path después de la reconfiguración es similar a los conmutadores estándar, pero en la topología de conmutadores estándar el aprendizaje y el reenvío de tramas está restringido por el protocolo de árbol de expansión (RSTP) a la topología activa de árbol, mientras que en ARP-Path se aprende y reenvía por todos los enlaces.

En STP, cuando un puerto ya no es parte de la topología activa debido a un fallo o tras ser inhabilitado por el operador, los hosts dejan de ser accesibles a través de ese puerto, por lo que todas las direcciones MAC asociadas a ese puerto se eliminan de la caché. Cuando un enlace se desactiva, las direcciones asociadas al puerto correspondiente se borran. Contrariamente a los conmutadores convencionales que utilizan STP, en los puentes ARP-Path no es necesario propagar los mensajes de notificación de cambio de topología (TCN) por toda la red, porque la gran mayoría de los caminos existentes siguen siendo válidos, todos los que no utilicen el enlace o conmutador en fallo. Los caminos que los utilizan se repararán (y las direcciones se reaprenderán) sólo cuando sea necesario, como se describe a continuación.

5.2.4 Reparación de camino

Los caminos seleccionados pueden deshacerse debido a la expiración del temporizador (de algún puerto en estado *confirmed*), por fallos de algún enlace (esto a su vez provoca el borrado de las direcciones MAC asociadas a los dos puertos del enlace), etc. Cuando sucede, la asociación <puerto, dirección MAC> pasa a estado (*repairing*) o en reparación. Para la reconstrucción de un camino se procede de la siguiente manera según el ejemplo de la Ilustración 63.

El conmutador que recibe una trama con destino desconocido, devuelve la trama unicast recibida encapsulada dentro de un mensaje de control denominado <*Path_Fail*>, al host origen de la misma, poniendo como destino la dirección MAC multicast *MCAST_All_Fast-Path_Bridges*, dirección especial de capa dos del grupo multicast asociada a todos los puentes ARP-Path. El conmutador con conexión directa a dicho host origen, intercepta la trama y reinicia la creación del camino ARP-Path mediante un mensaje de control <*Path_Request*>, que funciona de forma similar a un *ARP-Request*, inundando la red ARP-Path, reenviándose hasta encontrar el conmutador destino. Este mensaje <*Path_Request*> será interceptado por el conmutador conectado al host de destino que generará un mensaje de control <*Path_Confirm*> como respuesta que, recorriendo el camino bloqueado por el mensaje <*Path_Request*>, confirmará en cada uno de los conmutadores las direcciones origen y destino originales de la trama interna encapsulada.

La interceptación de mensajes de reparación por parte de los conmutadores frontera, aquellos conectados directamente a los hosts implicados en el camino, es posible porque los conmutadores frontera conocen las direcciones MAC de los hosts directamente conectados mediante una serie de mensajes <*HELLO*> que los conmutadores ARP-Path intercambian entre sí. Cuando un conmutador ARP-Path no recibe mensajes <*HELLO*> por un puerto, sabe que las direcciones MAC origen de las tramas recibidas por ese puerto corresponden a hosts a su cargo, conectados directamente, o a través de conmutadores estándar.

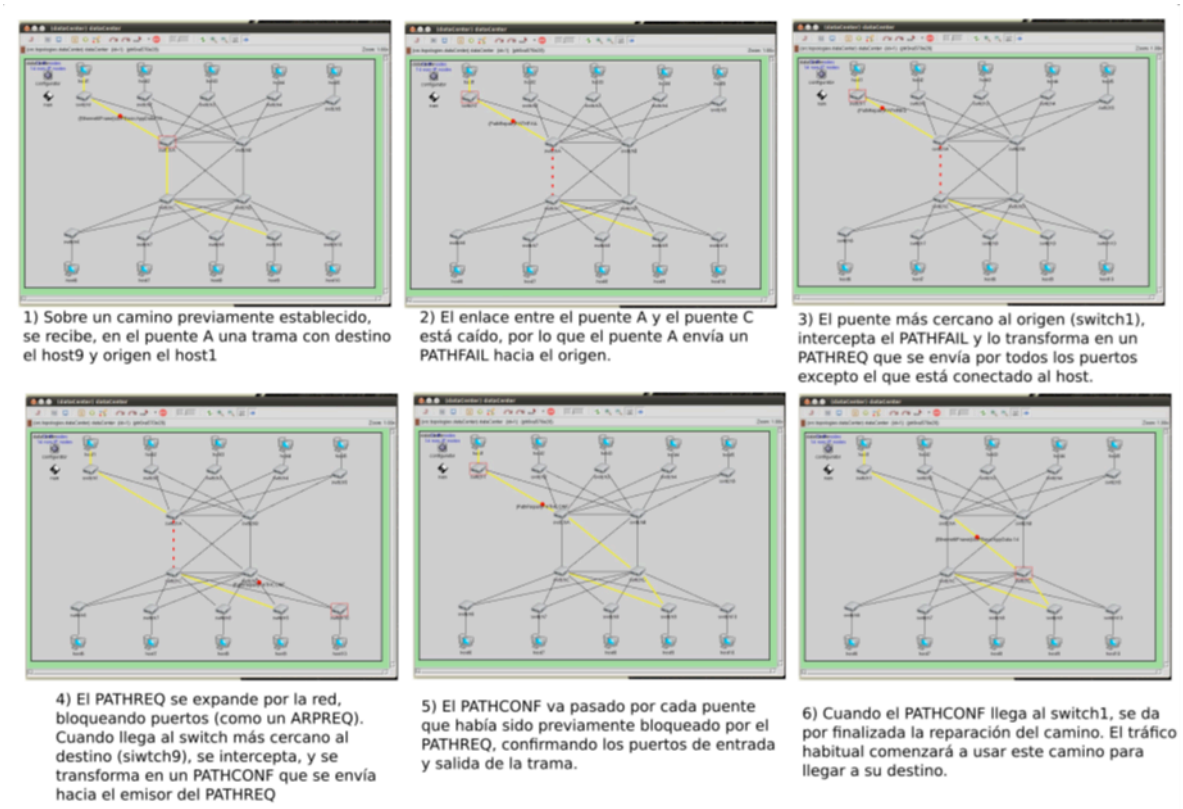


Ilustración 63. Reparación de camino (simulador Omnet)

Como alternativa, el camino puede ser reparado directamente a partir del puente afectado mediante la difusión, ya sea un mediante un *ARP-Request* estándar difundido en nombre del host origen, o mediante un mensaje <*Path_Request*> dirigido a la dirección de grupo multicast *MCAST_All_Fast_Path_Bridges*. En el primer caso, el *ARP-Request* es respondido por el host de destino con un *ARP-Reply* que selecciona la ruta hacia el conmutador que inició la reparación, el cual lo intercepta. En el segundo caso, un mensaje <*Path_Request*>, conteniendo las direcciones MAC e IP origen y destino, se transmite en la dirección hacia delante y es procesado y transmitido por todos los conmutadores atravesados hasta el conmutador conectado al host de destino. El pseudocódigo de la Ilustración 64 resume el procesamiento de tramas en un conmutador ARP-Path.

Una variante, desarrollada posteriormente y denominada reenvío inverso, de este procedimiento de reparación, opera de la siguiente manera: la trama unicast recibida en un conmutador que no tiene ruta para su dirección destino, es devuelta directamente sin cambios por el conmutador, por el puerto asociado a su dirección origen, en dirección contraria. El conmutador anterior en el camino, al recibirla por el puerto por donde suele enviarla, detecta que es una trama devuelta y la procesa con lógica invertida, cambiando la función del campo de dirección MAC origen por la de destino y viceversa. De esta forma la trama llega finalmente al conmutador frontera, asociado al host origen, el cual al detectar la trama devuelta genera un *ARP-Request* en nombre del host origen para rehacer el camino de nuevo.

```
Frame processing at bridge
-Destination address is broadcast or multicast
   - Destination address is multicast ARP-path: process as control
frame:
      -  Is Path Fail message: resend ARP Request with frame data)
   - Else if:
        - source address is unknown
           -Lock  temporarily source address to input port
        - source address is known (a ARP-path exists)
           -Discard frame if input port is not the associated port
 -Forward frame through all ports except prohibited turns and refresh
persistency timer of source address
-Destination address is unicast
    - Destination address is known
       -Frame is ARP Reply to a pending ARP Request:
          - Confirm locked address (frame destination address  to
            output port). Activate persistence timer.
          -  Associate source address of unicast frame to input
             port. Start refresh timer.
-  Else if Source address is known
              -Forward to associated output port. Refresh timers at
               ports for source and destination  addresses
      - Else:  associate source address of unicast frame to input
         port. Start refresh timer
  - Destination address  is unknown
        -Send Path Fail  in backward direction, encapsulating packet
header in multicast frame
```

Ilustración 64. Seudocódigo de procesado de tramas del protocolo ARP-Path

5.2.5 Máquina de estados de la asociación <puerto, dirección MAC>

Los conmutadores ARP-Path sustituyen el mecanismo estándar de aprendizaje de direcciones MAC (*backward learning*) por un nuevo mecanismo para la asociación de las direcciones aprendidas a los puertos del conmutador. La Ilustración 65 muestra su lógica mediante una máquina de estados finitos. Las elipses indican los estados de dirección y las transiciones muestran en la línea superior el evento (tipo de mensaje recibido en cursiva) y en la línea inferior (en negrita) la acción realizada.

Una dirección MAC se encuentra en estado liberado *(released)* cuando no está asociada a ningún puerto del conmutador (es decir, que le es desconocida). Cuando una petición *ARP-Request*, con origen en la dirección MAC *A*, se recibe en un conmutador ARP-Path a través de su puerto *x*, la dirección MAC *A* cambia su estado a *locked* durante la duración del temporizador *lock_To,* o hasta que la correspondiente respuesta ARP (*ARP-Reply*) sea recibida, lo que suceda primero. Si la respuesta se recibe antes de la expiración del temporizador *lock_To*, el estado cambia a aprendido/confirmado (*learned/confirmed*), por lo que la asociación de la dirección al puerto *x* es ahora firme; de lo contrario la asociación expira y el estado vuelve a liberado (*released*). El estado aprendido o confirmado se mantiene al menos durante un período del temporizador *confirmed_To*. Este temporizador es similar a los temporizadores de caducidad de caché (300 s) de un conmutador estándar y se renueva al recibirse tramas unicast de esa dirección por el puerto asociado.

Una dirección en estado liberado (desconocida para el puente) pasa directamente a estado aprendido/confirmado cuando se recibe un mensaje de respuesta *ARP-Reply*, de *B* a *A*, a través del puerto *y*, quedando además asociada la dirección *B* al puerto *y*.

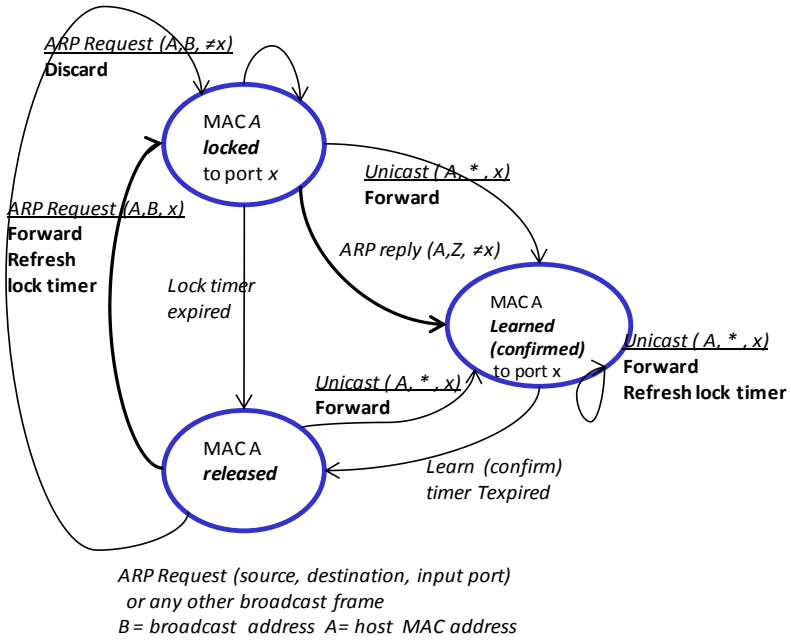


Ilustración 65. Máquina de estados de la asociación <puerto, MAC> (bloqueo y confirmación)

5.2.6 Distribución de carga

Los conmutadores ARP-Path establecen caminos bajo demanda, lo que significa que las rutas obtenidas se adaptan a las condiciones de carga de la red de cada momento sin necesidad de utilizar ningún mecanismo de balanceo de carga. Por diseño, cuando se solicita un nuevo camino, el camino más rápido para llegar al host de destino será el seleccionado. Esto significa que los nuevos caminos seleccionados hacia hosts (distintos de los hosts activos existentes comunicantes) seguirán el camino más rápido en el momento de la solicitud, por lo tanto, las rutas con mayor carga (retardo) no serán seleccionadas si existen rutas alternativas con menor retardo. En su lugar, tenderán a ser seleccionados los conmutadores con latencia más corta. Debido al elevado número de hosts, como el camino establecido tiene granularidad por cada host, el tráfico quedará equilibrado en toda la infraestructura, lo que resulta en latencias menores. Esta funcionalidad nativa del protocolo ARP-Path se estudia en detalle en la segunda parte de este capítulo, dedicada al simulador de flujos.

5.2.7 Compatibilidad con puentes estándar y enrutadores

Los conmutadores ARP-Path pueden operar conectados a puentes estándar 802.1D en modo de núcleo-islas, como se muestra en la Ilustración 66. Un núcleo de conmutadores ARP-Path puede interconectar islas de puentes estándar, que ejecutan el protocolo de árbol de expansión RSTP internamente y en su conexión al núcleo central. La autoconfiguración de las islas de puentes estándar funciona como sigue: los conmutadores ARP-Path conectados a los puentes estándar reciben las BPDU estándar en los puertos de conexión a las islas de puentes estándar. Como consecuencia de ello se ejecuta el protocolo estándar RSTP en esos puertos, emitiendo BPDUs para anunciar al conmutador ARP-Path como si tuviera una conexión directa con un puente raíz (virtual, materializado en el núcleo de conmutación ARP-Path) con prioridad máxima de puente. Por lo tanto, los conmutadores ARP-Path serán seleccionados automáticamente como puentes raíz por los puentes estándar conectados (con enlaces punto a punto) a ellos, colgando de cada uno de ellos uno o varios árboles completamente separados de los árboles vecinos. Los enlaces que unen a dos árboles serán

bloqueados por el protocolo RSTP al ser detectados como de mayor coste al puente raíz virtual del núcleo. De esta manera se crean una serie de árboles disjuntos, todos enraizados en el núcleo ARP-Path. A diferencia de otras propuestas, no se precisa encapsular ni modificar las tramas insertando etiquetas para atravesar el núcleo y enviar tramas de una isla a otra.

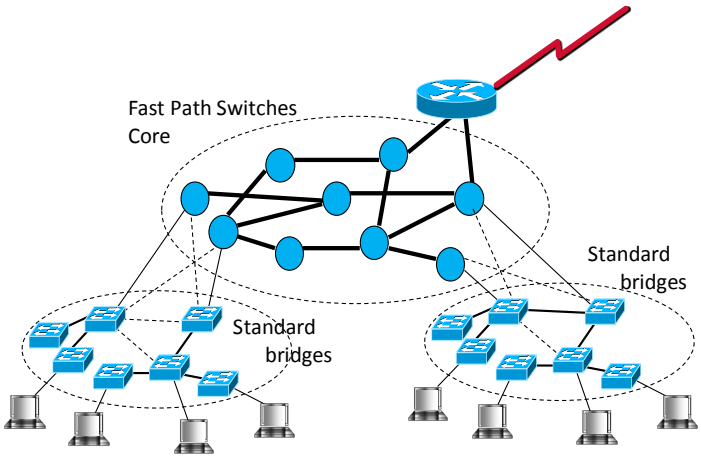


Ilustración 66. Red híbrida núcleo-islas de puentes ARP-Path y puentes estándar IEEE 802.1D

5.2.8 *Etherproxy* en conmutadores frontera

Hay dos problemas principales a resolver en la escalabilidad de Ethernet: el encaminamiento eficiente en capa dos Ethernet (en sustitución del protocolo *Spanning Tree* [Per85] para superar sus limitaciones en el numero de enlaces activos y de tamaño de red), y la reducción del número de tramas difundidas a toda la red (*broadcast*). El uso del protocolo ARP [RFC826] en redes conmutadas de gran tamaño es criticado frecuentemente por el aumento de la carga de proceso en los hosts para resolver las peticiones ARP cuando crece el número de hosts de la red, ya que, como se ha comentado antes, todos los hosts de la red deben procesar los *ARP-Request* recibidos y su carga aumenta innecesariamente [MEZ04]. ARP-Path se centra en la obtención de rutas óptimas y Etherproxy [EC09] en la reducción de las tramas de difusión.

Hay dos aspectos de la difusión que son de interés: difusión entre los puentes y difusión a los hosts.

Etherproxy es una propuesta reciente que se puede combinar con ARP-Path, ya que se centra en reducir al mínimo las emisiones para aumentar la escalabilidad de Ethernet a límites más altos (por ejemplo, 50K hosts) que impondrían una carga significativa en los hosts al tener que procesar cada host todas las solicitudes ARP inundadas. Etherproxy minimiza el tráfico de difusión en la red mediante almacenamiento en caché de los pares de direcciones <IP-MAC> aprendidas de los paquetes de respuesta *ARP-Reply*, respondiendo directamente a las peticiones *ARP-Request* de las que tiene respuesta válida en su caché ARP. La ubicación recomendada de los dispositivos Etherproxy está en el borde de la red, con un Etherproxy sirviendo hasta 500 hosts. Etherproxy se puede implementar ya sea como un dispositivo independiente o en el interior de los conmutadores.

La implementación del *proxy* ARP dentro de los conmutadores frontera ARP-Path presenta ventajas de sinergia, dado que capacitar a un conmutador ARP-Path para actuar como *proxy* ARP requiere añadir únicamente una columna <dirección IP> a las tablas de direcciones MAC aprendidas por el conmutador. Su posición en la frontera de la red previene completamente la inundación y en

caso de fallo de la consulta a la tabla caché, el *ARP-Request* es reenviado de la forma estándar y el proceso de establecimiento del camino es el descrito anteriormente.

La tasa efectiva de solicitudes ARP con el uso de cachés queda reducida por la tasa de fallo de caché (*miss rate, m)* , donde $m = 1 - h$ (siendo h la tasa de aciertos de la cache)

$$\textit{Tasa efectiva (ARP-Request)} = \textit{Tasa(ARP-Request)} \cdot (1-h) \qquad (5.1)$$

Como se muestra en [EC09] la tasa de aciertos puede llegar al 100% si los conmutadores realizan un refresco proactivo de las direcciones de hosts próximas a expirar en su caché. Este refresco consiste en un *ARP-Request* 'unicast' enviado por el *proxy* al host con la dirección MAC próxima a expirar y el correspondiente *ARP-Reply* generado como respuesta por el host, que refresca la caché. Sin refresco, la tasa de éxito del proxy se sitúa entre 0,6 y 0,8 dependiendo del tipo de red. Entre los costes de Etherproxy figuran los siguientes: mayor procesado en los puentes de las peticiones ARP mediante el envío de respuesta ARP a los solicitantes y mayor anchura de los registros de las cachés. Etherproxy es autoconfigurable, como ARP-Path, por lo que la combinación de ambos lo es también.

5.3 Evaluación

Para la evaluación del protocolo ARP-Path se han tenido en cuento distintos aspectos como la complejidad computacional, la cantidad de información de estado que necesita almacenar y el grado de utilización de la infraestructura de la red. La evaluación se ha basado fundamentalmente en el uso de simuladores, pero también se incluyen algunas medidas obtenidas mediante una implementación realizada en Linux con el objetivo de servir de 'prueba de concepto'.

5.3.1 Complejidad

Analizamos primero la complejidad computacional, de mensajes y de estado computacional de los conmutadores ARP-Path frente a los puentes transparentes que utilizan el protocolo de árbol de expansión y protocolos basados en enrutamiento de tipo 'estado de los enlaces'.

Tanto el protocolo de árbol de expansión como los protocolos de enrutamiento como IS-IS son proactivos: en el protocolo de árbol de expansión, cada nodo emite periódicamente su mejor BPDU (ruta de menor coste al puente raíz) a sus vecinos y procesa las d (siendo d el grado medio de la red) BPDUs recibidas de ellos, para seleccionar el vecino que ofrece la distancia más corta hacia el puente raíz como su puente principal, lo que significa que la complejidad del mensaje es $\Theta(d)$. Los conmutadores *Shortest Path* (utilizan el algoritmo de Dijkstra para el cálculo de caminos mínimos) tienen, para una red con N puentes, una complejidad de orden $\Theta(N^2)$ (reducible a $N \cdot logN$). Además de esto, es necesario un mecanismo adicional de sincronización para evitar bucles causados por inconsistencias transitorias de las tablas. Cada puente debe mantener informados a los demás puentes de los hosts conectados a él. Esto significa que las tablas de reenvío pueden llegar a ser muy grandes y la cantidad de tráfico de control importante para mantener actualizada en cada conmutador la lista de hosts activos y su conmutador asociado.

ARP-Path por el contrario es un protocolo reactivo, los conmutadores no intercambian información de enrutamiento de forma periódica. El mensaje *ARP-Request* estándar y su respuesta *ARP-Reply*, que se utilizan para marcar los caminos en la red, no tienen ningún coste adicional en cuanto a mensajes, con la excepción de las tramas que llegan a un conmutador con una ruta expirada,

típicamente después de un fallo de un enlace o nodo. Cuando esto sucede, los mensajes adicionales (ruta de solicitud/ruta de respuesta estándar o petición/respuesta ARP) se generan para reconstruir la ruta de acceso.

En cuanto al estado almacenado, los conmutadores ARP-Path guardan una cantidad de información de estado similar a la de los conmutadores estándar. Estos aprenden las direcciones MAC origen de los hosts activos asociando cada dirección a un puerto del conmutador durante un tiempo controlado por un temporizador de caducidad de la caché. Los puentes ARP-Path almacenan la misma asociación de direcciones MAC de cada puerto, pero utilizan dos temporizadores diferentes: uno para el bloqueo (de corta duración, para descartar las tramas duplicadas) y otro para el estado de aprendizaje (de larga duración) para la asociación de una dirección a un puerto. Así, sólo se requiere un temporizador adicional corto durante la fase de establecimiento de camino. El procesado de la expiración de direcciones para el período más corto (bloqueo) supone algún esfuerzo computacional adicional para el puente. El procesado del temporizador de aprendizaje (largo) es igual a los puentes estándar (con el mismo tiempo de espera, por defecto 300 segundos), pero el número de direcciones aprendidas con temporizadores a manejar es mucho menor. Hay que tener en cuenta que en ARP-Path, a diferencia de los puentes transparentes estándar, los conmutadores no realizan inundaciones de las tramas unicast cuando desconocen la dirección MAC destino.

5.3.2 Utilización de la infraestructura

En esta sección se compara el número de enlaces activos al usar ARP-Path y el protocolo RSTP respectivamente. ARP-Path no bloquea ningún enlace, por lo que los L enlaces de la topología están activos y pueden ser utilizados. RSTP solamente deja activos $(N-1)$ enlaces, que forman el árbol de expansión, y bloquea el resto para impedir los bucles. La relación de los enlaces activos en ARP-Path frente a RSTP es entonces:

$$U = L/(N-1) \qquad (5.2)$$

Sustituyendo $L = N*d/2$ en (5.2):

$$U = N*d/((N-1)*2) \approx d/2 \qquad (5.3)$$

La Tabla 62 muestra el rango de la relación de utilización de la infraestructura en redes de grado medio 4, 6 y 8. La más alta (a la izquierda) se corresponde con valores de redes de 16 nodos mientras que los valores más bajos (más a la derecha) del rango corresponden a valores crecientes de N (hasta 256). La mejora en la relación de enlaces activos para ARP-Path oscila entre 2 y 4,3 veces para promedio de grados de nodo de red de 4 a 8, respectivamente.

Tabla 62. Relación de enlaces activos ARP-Path respecto a STP

Grado medio del nodo	***d***	**4**	**6**	**8**
Relación de enlaces activos	*U*	2,1-2,0	3,2-3,0	4,3-4,0

En las secciones siguientes se realiza una evaluación más detallada de la utilización instantánea de los enlaces activos, localizando el enlace más cargado de la red en cada caso, el enlace que determina el rendimiento máximo de la red.

5.3.3 Validación

El protocolo ARP-Path fue validado mediante una implementación de tipo *prueba-de-concepto* en Linux (kernel 2.6), realizada en espacio de usuario (Bart de Schuymer [ISN+11]), por lo que se sacrificaron las prestaciones a la simplicidad de la implementación.

El esquema de la red utilizada se muestra en la Ilustración 67, consta de dos hosts que se conectan a la red corporativa UAH a través de tres conmutadores ARP-Path conectados en triángulo entre ellos. Las máquinas Linux son tarjetas Soekris Net5501 a 433 MHz con cuatro interfaces Ethernet a 100 Mbps. El montaje global puede verse en la fotografía de la Ilustración 68.

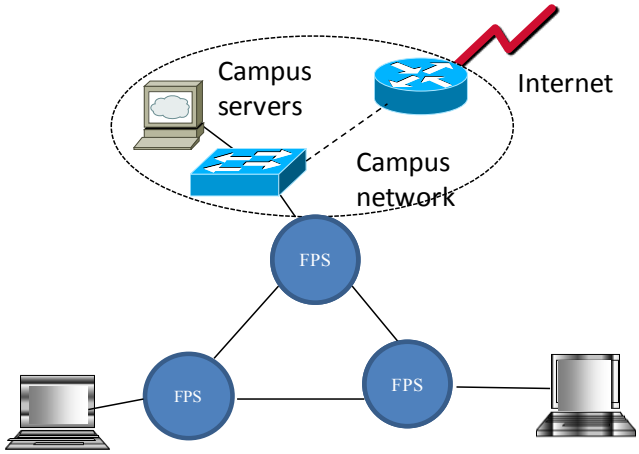


Ilustración 67. Red de validación del protocolo ARP-Path (esquema)

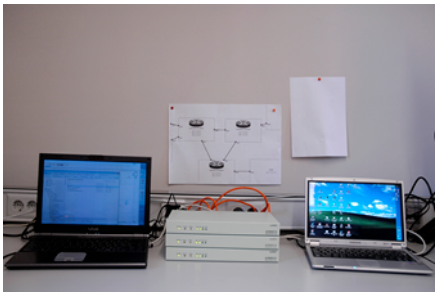


Ilustración 68. Red de validación protocolo ARP-Path (fotografía)

En los experimentos realizados, los hosts obtienen sus direcciones con normalidad (a través de DHCP [Dro97]) sin necesidad de configuración o modificación alguna. El acceso a páginas web, transferencia de archivos y reproducción de vídeo funciona con normalidad. No se producen tramas de difusión incluso uniendo dos puertos del mismo conmutador. Sin embargo, los bucles aparecen cuando el conmutador frontera de campus (puente estándar) se conecta a través de dos o más enlaces de un puente estándar a la red de puentes ARP-Path, al no existir activo un mecanismo común de prevención de bucles. Las condiciones de compatibilidad con puentes estándar con árbol de expansión se describen más adelante.

Se midieron y compararon los retardos de *ping* entre dos hosts separados por un puente ARP-Path basado en Linux con los de un conmutador Ethernet estándar D-link de 10/100 Mbps con la misma conectividad. El modo auto-negociación se estableció en ambos casos. Cuando el host origen carece de la MAC del host destino en su caché, emite una solicitud ARP. El primer *ping* puede tardar hasta 48 ms en ARP-Path (Linux) y sólo 2,43 ms en el puente hardware estándar, ya que es necesario

hacer una transferencia entre el núcleo y el espacio de usuario para establecer la ruta de acceso en nuestra implementación Linux. Una vez que la ruta se ha establecido con el primer par de solicitud y respuesta ARP, la respuesta al *ping* sólo tarda 238 μs de media para el puente Linux ARP-Path y 200 μs para el puente de hardware estándar. La razón de la alta velocidad de conmutación de Linux cuando se trata de una dirección que ya ha aprendido, es porque en este caso el reenvío se realiza directamente por el núcleo, sin necesidad de cambiar al espacio de usuario.

También se ha validado la reparación de caminos en la red de la Ilustración 67. Mientras se envían *pings* continuamente de un host a otro a través del camino establecido entre los dos hosts, a través del enlace directo que los une, se desconecta dicho cable. Una vez se detecta el fallo del enlace Ethernet, el tiempo de restablecimiento del camino es de 672 ms, medidos desde el ultimo *ping* enviado. Estableciendo un árbol de expansión en la red, los dos hosts quedan a una distancia de dos saltos debido al bloqueo del enlace redundante directo que los une. El retardo con ARP-Path se duplica ahora hasta los 880 μs, como era de esperar. La tasa máxima de solicitudes ARP que soporta la implementación en Linux es de 100 *ARP-Request*/segundo.

5.3.4 Simulaciones

Se ha implementado un simulador de ARP-Path en OMNET++ [OMNET] utilizando la *suite* de protocolo INET, para ello hubo que modificar la implementación de un conmutador Ethernet estándar. Se compararon las prestaciones del protocolo ARP-Path con los protocolos *Shortest Path* y con el protocolo STP, centrándonos en el estudio del caudal máximo resultante con cada mecanismo de reenvío, no en la dinámica de la creación de los caminos. Se usaron dos topologías de red: una red empresarial genérica de dos niveles (Ilustración 69) y una red paneuropea de referencia [NRS].

5.3.4.1 Red empresarial

Para obtener medidas de latencia, simulamos sesiones TCP de datos de tamaño creciente (2 KB, 100 KB y 100 MB de información) entre 3 hosts remitente y receptor (3 hosts marcados como 1, 2, 3 y 4, 5, 6, respectivamente, en la Ilustración 69). Los experimentos se repitieron 100 veces. Todos los enlaces de la red tienen un retardo de propagación de 1 μs y una velocidad de transmisión de 100 Mbps. Cuando la red está poco cargada, la latencia de red desde el host origen hasta el host destino (medido desde la capa de Ethernet de la fuente a la capa de Ethernet de destino) es igual a 275 μs, en promedio para los tres protocolos con pequeñas variaciones. Para cargas medias (sesiones de 100 KB de información), la latencia de la red llega a 370 μs, en promedio, para los tres protocolos. Y con cargas elevadas (sesiones de 100 MB de información), la latencia se mantiene moderada para *routers* y conmutadores ARP-Path, pero crece a 1645 μs para el árbol de expansión, debido a la congestión de los enlaces compartidos por muchos caminos (enlaces alrededor de la raíz). Por lo que llegamos a la conclusión de que la latencia de ARP-Path es equivalente a la de enrutamiento *Shortest Path* y ambas son bastante inferiores al árbol de expansión con cargas de red elevadas debido a la congestión en los enlaces de los niveles superiores del árbol.

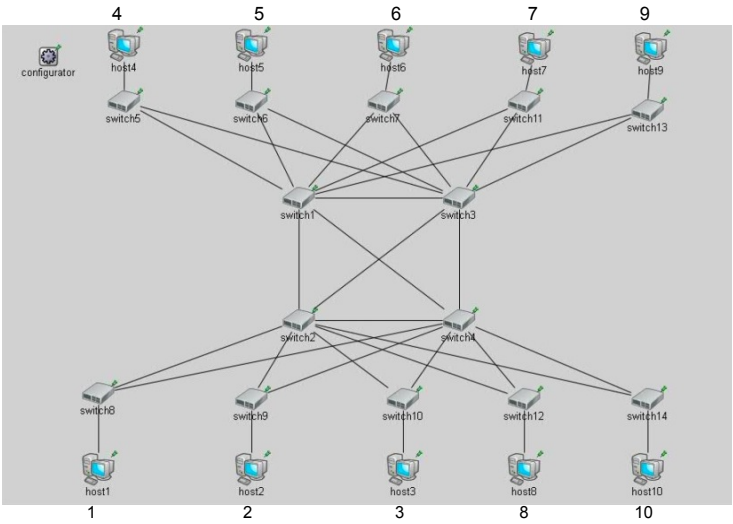


Ilustración 69. Red empresarial, topología activa (salvo STP)

Para comparar el rendimiento de los tres protocolos en una red modelo de centro de datos se optó por una topología como la de la Ilustración 69. Se realizaron simulaciones en las que los hosts situados en la parte inferior establecían sesiones de intercambio de datos con hosts situados en la parte superior (por lo que los flujos de datos debían atravesar los enlaces del núcleo central de la topología). Se establecieron pares de hosts, de 1 a 9, de 2 a 7, de 3 a 5 y de 8 a 4 predeterminados par el establecimiento de sesiones. Los flujos de datos consistían en sesiones UDP con tamaños de datos variables (desde 1 a 60 KB de datos enviados cada 150 ms, en media, siguiendo una distribución exponencial). Las ejecuciones tenían una duración de 3 s.

La Ilustración 70 muestra los resultados obtenidos para el enlace más cargado de la topología (enlace del núcleo) en función de la cantidad de tráfico inyectado desde cada hosts (representado también como porcentaje de uso del propio enlace de acceso de cada hosts). Se observa como a medida que aumentamos la cantidad de tráfico que debe cruzar el núcleo, va aumentando la utilización de esos enlaces hasta alcanzar eventualmente su saturación.

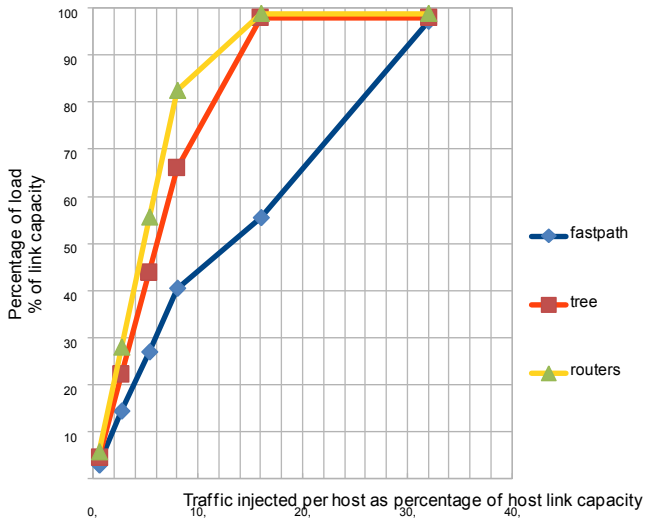


Ilustración 70. Comparación de rendimiento. Porcentaje de carga en enlace más cargado respecto al tráfico inyectado (en porcentaje de la capacidad del enlace del host origen)

ARP-Path presenta una característica de saturación mucho más retrasada debido a que distribuye los flujos entre los distintos caminos posibles para cruzar el núcleo, presenta un fenómeno de diversidad de caminos muy favorable debido a la multiplicidad de caminos de igual coste disponibles en la topología. *Shortest Path* necesitaría estrategias multicamino como ECMP para poder aprovechar esa ventaja mientras que STP simplemente la ignora y selecciona un único enlace de los cuatro disponibles para atravesar el núcleo. Numéricamente, ARP-Path absorbe hasta el 32% de la capacidad de los enlace de hosts antes de saturar, mientras que los otros dos protocolos lo hacen a la mitad de carga (16%).

5.3.4.2 Red Paneuropea

También se simuló el núcleo de la red pan-europea, una malla plana de 16 nodos [NRS]. En este caso se envía tráfico UDP con sesiones de 1 a 60 KB de tamaño y tiempo medio entre mensajes con distribución exponencial de promedio 75 ms. Los retardos de propagación de los enlaces se asignaron proporcionalmente a la distancia real sobre el mapa de la ubicación de cada nodo con valores de entre 1 y más de 3 ms. Todos los enlaces tienen la misma capacidad. El tráfico se origina en los ordenadores de los nodos al oeste y se dirige hacia los del este para facilitar la saturación de los enlaces. La activación de tráfico en los nodos se secuencia al azar con un retraso medio de 0,5 segundos entre activaciones. Como se muestra en la Ilustración 71, el enlace más cargado satura un poco más tarde para ARP-Path que con enrutadores de camino más corto y mucho más tarde que con el árbol de expansión.

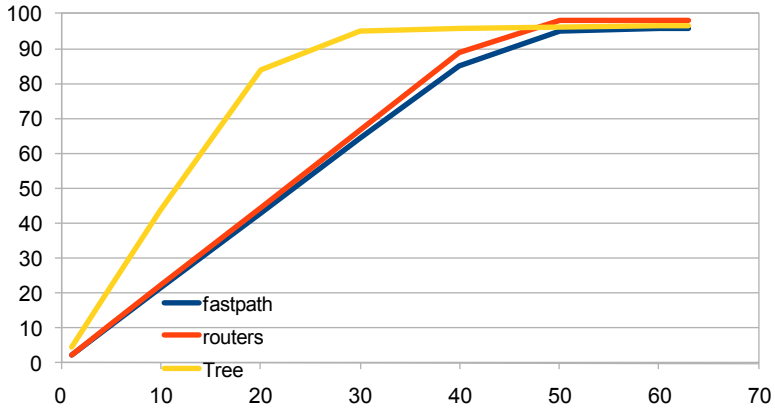


Ilustración 71. Comparación de rendimiento de red pan europea en % de utilización del enlace más cargado respecto a % de carga media aplicada en los enlaces de host

5.4 Simulador de flujos para ARP-Path

En esta sección se describe el simulador de flujos desarrollado para caracterizar protocolos avanzados de redes, entre ellas las redes de conmutadores ARP-Path. Tras un resumen de los modelos utilizados para representar el comportamiento del tráfico y del protocolo estudiado, se muestran los resultados obtenidos con dichos modelos en diversas topologías y las conclusiones relativas a las prestaciones del protocolo ARP-Path y su validación en otros simuladores como Omnet++ y algunas pruebas experimentales ya comentados más arriba.

5.4.1 Introducción

Tradicionalmente se han utilizado simuladores de paquetes para caracterizar el comportamiento de protocolos y redes de comunicaciones mediante simuladores de eventos discretos. Sin embargo, el coste computacional de modelar individualmente cada paquete de cada flujo de datos en una red de tamaño medio/grande (cientos o miles de nodos y flujos entre ellos) y obtener resultados de su comportamiento en estado estacionario es prohibitivo, tanto en términos de capacidad de proceso como de memoria requerida para soportarlo.

Un enfoque diferente cuyo objetivo es reducir drásticamente los recursos necesarios para su implementación consiste en modelar los intercambios de datos con un mayor nivel de abstracción, trabajando con flujos de datos caracterizados por una determinada duración y tasa media de transferencia en lugar modelar cada paquete de manera individualizada.

Algunos trabajos como Bertsekas [BG92] justifican el uso de simuladores de flujo si se cumplen determinadas condiciones como que el número de flujos para cada par origen/destino dentro de la red sea grande y que la tasa media de cada uno sea pequeña en relación a la capacidad de los enlaces de la red.

El coste de utilizar este mayor nivel de abstracción en el modelado del tráfico es pequeño, si el objetivo del estudio en cuestión es caracterizar comportamientos globales dentro de la red como pueden ser la distribución macroscópica de la carga dentro de la red o el tamaño medio de las rutas seleccionadas por un determinado protocolo de encaminamiento. A cambio, permite evaluar multitud de topologías, de todo tipo y tamaño, tanto reales como virtuales (generadas mediante herramientas automáticas).

El objetivo fundamental del simulador de flujos que se presenta en este capítulo es evaluar la capacidad de distribución de carga del protocolo ARP-Path. ARP-Path es un protocolo de encaminamiento entre conmutadores avanzados que explora, bajo demanda, todos los caminos disponibles para alcanzar un destino determinado y se queda con aquel que proporciona una menor latencia en el momento de la exploración. Por lo tanto, se trata de un protocolo sensible al retardo (y por extensión a la carga) de la red.

Dado que el simulador no incorpora la capacidad de manejar paquetes individuales de datos que permitan modelar el comportamiento real del proceso de exploración de caminos (mediante una difusión controlada), resulta especialmente importante cómo se implementa la capacidad de modelar el retardo esperado en el tránsito de un paquete por los distintos enlaces de la red, en función del estado de carga de cada uno de los enlaces en el momento de la exploración. En este caso, se ha optado por representar la latencia en la red mediante un modelo de costes por enlace, variables con la carga, que permite aplicar un algoritmo clásico de camino más corto como Dijkstra para la selección de la ruta asignada a cada flujo.

Por otra parte, el modelado del tráfico de datos que circulan por la red, tanto en lo que respecta al conjunto de nodos origen/destino de los mismos como a la cantidad de información que transportan, requiere la incorporación de los denominados modelos de gravedad y modelos de flujos de datos adaptados a distintas condiciones de funcionamiento de la red (extraídas de trazas reales o generadas aleatoriamente). Los modelos de gravedad permiten representar el peso relativo de cada nodo de la red en la cantidad total de tráfico que la atraviesa; permiten definir cuántos flujos tienen a cada uno de los nodos de la red como origen y/o destino del mismo. Los modelos de flujos permiten caracterizar los flujos individuales mediante una tasa de transferencia (velocidad binaria) y una duración (volumen total de información transportada).

El simulador se ha desarrollado en Python utilizando las librerías de simulación Simpy [SIMPY].

Las siguientes secciones presentan y justifican los modelos de latencia (costes) y tráfico (modelos de flujo y modelos de gravedad) que se implementan.

5.4.2 Modelo de costes (latencia)

Si el número de flujos de datos activos en la red es grande podemos asumir que el retardo que sufrirá un paquete para atravesar un enlace es independiente del comportamiento del resto de enlaces de la red y, por tanto, la latencia total acumulada entre los nodos origen y destino del flujo puede calcularse como la suma de los retardos acumulados en cada de los enlaces que forman el camino seleccionado.

Sea $N \equiv \{A,B,\ldots,Z\}$ el conjunto de nodos de la red. Sea P_{IJ} el conjunto de caminos posibles entre dos nodos I y J cualesquiera en la red ($I,J \in N$) y $L \equiv \{L_1, L_2,\ldots,L_m\}$, el conjunto de enlaces de la red. Sea $Pi \equiv \{L_1, L_2,\ldots,L_k\} \in P_{IJ}$ uno de los posibles caminos entre los nodos I y J donde ($L_1, L_2,\ldots,L_k \in L$).

La latencia esperada en el camino P_i ($L(P_i)$) puede calcularse como la suma de las latencias esperadas en cada uno de los enlaces que lo forman:

$$L(P_i) = \sum L(L_j),\ \forall L_j \in P_i \qquad (5.4)$$

La latencia esperada por paquete en un enlace se calcula como la suma de los retardos producidos por el proceso previo a la transmisión en cada nodo (T_p), la propagación de la señal física en el medio (t_r), el tiempo de transmisión (X_p) de los bits que componen la trama en curso y el tiempo de espera en cola.

$$L(L_j) = T_p + t_r + X_p + E[w] = T_p + t_r + E[t] \qquad (5.5)$$

En un entorno clásico Ethernet los tiempos de proceso en los conmutadores se encuentran en el rango de 1-2 μs [CIS11] [RFC1242], una trama típica (que transporte un mensaje *ARP Request* ocupa unos 64 octetos y los tiempos de propagación varían entre 5 ns y 5 μs (para enlaces de entre 1m y 1 km de longitud). La siguiente tabla muestra valores de latencia total mínima (sin tiempo de espera en cola) típicos para distintas configuraciones o escenarios:

Tabla 63. Latencia total por enlace para trama mínima

R	***t_r***	***X_p (64 octetos)***	***Tp***	**Suma**
100 Mbps	5 ns-5μs	5,12 μs	1-2 μs	~ 6-12 μs
1 Gbps	5 ns-5μs	512 ns	1-2 μs	~ 1,5-7,5 μs
10 Gbps	5 ns-5μs	51,2 ns	1-2 μs	~ 1-7 μs

Por su parte, los tiempos de espera en cola dependen fundamentalmente de la carga del enlace. En condiciones de independencia entre enlaces (tal como se comentó anteriormente) podemos aplicar alguno de los modelos clásicos de teoría de colas [Kle75] para estimar el tiempo medio de espera en cola. En concreto, el modelo markoviano puro M/M/1 nos proporciona una cota máxima de dicho valor (asume distribuciones poissonianas tanto en la generación del tráfico como en el tiempo de transmisión (tamaño de las tramas exponencial)).

Las expresiones del tiempo de espera en cola y del tiempo de tránsito total (espera + transmisión) de un sistema clásico M/M/1 son:

$$E[w] = (1/\mu)*\rho/(1-\rho) \qquad (5.6)$$
$$E[t] = (1/\mu)/(1-\rho) \qquad (5.7)$$

Si consideramos tramas Ethernet de tamaño máximo (1500 octetos) y un valor de R = 100 Mbps, el tiempo medio de transmisión de trama será Xp = 120 μs. La gráfica siguiente muestra el tiempo de tránsito medio y el tiempo medio de espera en cola (tienen un desfase inicial de $Xp = 120$ μs).

Cabe destacar sobre la gráfica que para un valor de carga del 50% ($\rho=0,5$), los tiempos de transmisión y espera se igualan, mientras que para factores de carga muy altos (por ejemplo para $\rho=0,99$) el tiempo de espera es 100 veces mayor que X_p y, por tanto, el factor dominante en el retardo total esperado en el enlace.

La ecuación (5.5) puede reescribirse interpretando que la latencia total tiene una parte fija (tiempos de proceso, transmisión y propagación) y una parte variable dependiente de la carga (espera):

$$L(L_j) = T_F + E[w] \qquad (5.8)$$

En igualdad de condiciones respecto a la parte fija de dos enlaces (longitud y régimen binario), la diferencia de latencia entre ambos vendrá determinada por la carga relativa de cada uno. De la misma manera, dada una red con enlaces homogéneos (parte T_F similar), la diferencia de latencia entre dos rutas con igual número de saltos vendrá también determinada por el estado de carga de cada uno de los enlaces que componen cada ruta.

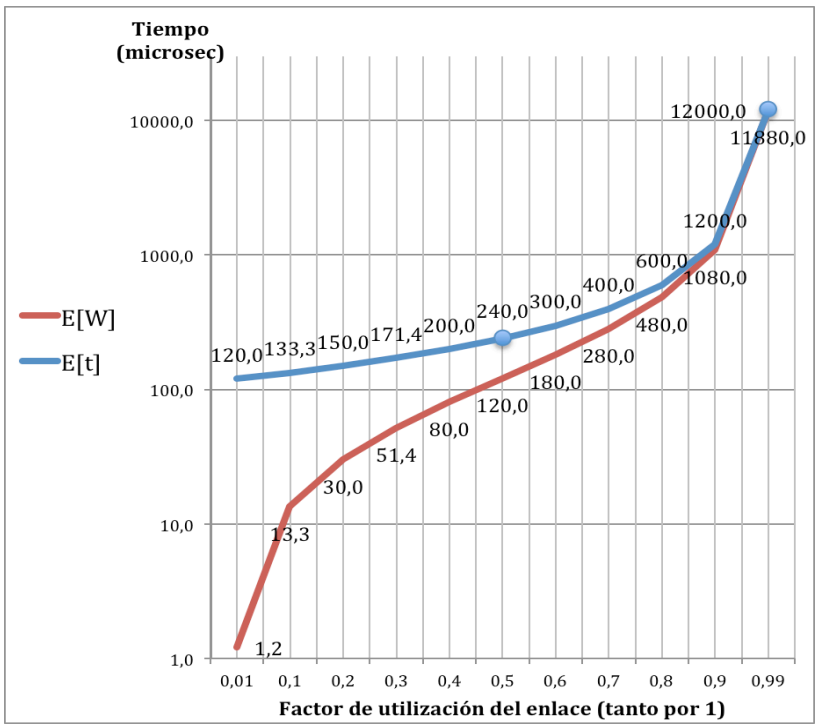


Ilustración 72. Representación de los tiempos medios de tránsito E[t] y espera en cola E[w] de en función del factor de carga r de un sistema clásico M/M/1

Las siguientes tablas muestran algunos valores característicos de L para el caso de topologías específicas de este estudio como:

Un centro de datos (DC):

Tabla 64. Valores característicos típicos de un centro de datos

R	***t_r (d=1m)***	***T_p***	***X_p < 1500 octetos***	**T_F**
1-10 Gbps	5 ns	2 μs	1-10 μs	~ 3-12 μs

Tabla 65. Valores de retardo de espera en cola y latencia total por enlace en función de la carga, en un centro de datos

ρ_{Li}	***E[w] (μs)***	***L_{Li} (μs)***
0	0	~ 3-12
0,2	0,3-3	~ 3,5-12,5
0,4	0,8-8	~ 4-20
0,6	1,8-18	~ 5-30
0,8	4,8-48	~ 8-60

De los valores de la tabla podemos concluir que el tiempo de propagación es irrelevante en el cómputo total. En cambio, la parte fija (T_F) de la latencia es determinante de su valor absoluto salvo para valores de carga altos (40% y superiores). Por tanto, al computar la latencia comparada entre dos caminos para elegir el de menor retardo, el sistema exhibirá una cierta inercia a seleccionar caminos con menor número de saltos salvo situaciones de carga descompensada.

Una red campus o metro:

Tabla 66. Valores característicos típicos de una red campus o metro Ethernet

R	***t_r (100-1000m)***	***T_p***	***X_p < 1500 octetos***	**T_F**
100-1000 Mbps	0,5-5 μs	2 μs	10-100 μs	~ 10-100 μs

Tabla 67. Valores de retardo de espera cola y latencia total por enlace en función de la carga, en una red campus o metro Ethernet

ρ_{Li}	***E[w] (μs)***	***L_{Li} (μs)***
0	0	~ 10-100
0,2	3-30	~ 15-130
0,4	8-80	~ 20-180
0,6	18-180	~ 30-280
0,8	48-480	~ 60-580

En este caso, al disminuir la velocidad de los enlaces aumentan los tiempos de transmisión y, en consecuencia, también los de espera. La variación en el tiempo de propagación debido a la longitud de los enlaces tiene un efecto pequeño en la latencia total y es completamente despreciable en cargas medias y altas. El tiempo de proceso tiene un impacto pequeño en la latencia total, despreciable salvo en cargas muy bajas. En la comparación de latencia acumulada en distintas rutas, el efecto de la carga (tiempo de espera) es más acusado que en el caso de los centros de datos por lo que se seleccionarán caminos más largos pero más descargados con mayor probabilidad.

Una red de interconexión (POP):

Las redes de interconexión presentan un grado de heterogeneidad mucho mayor que la observada en los casos anteriores, tanto la longitud (propagación) como la velocidad de las líneas (transmisión y espera) pueden presentar variaciones de hasta dos órdenes de magnitud o incluso más. La única conclusión no arriesgada que puede extraerse en este caso es la escasa influencia del tiempo de proceso salvo para enlaces de muy alta tasa binaria (10 Gbps).

Dado que nuestro simulador de flujos no contempla la posibilidad de modelar paquetes individuales con los que medir la latencia en todos los posibles caminos para seleccionar el mejor posible (por el sencillo método de la difusión tal como hace el protocolo ARP-Path), debemos diseñar un procedimiento de selección de rutas que tenga en cuenta todos los factores mencionados en la discusión previa y permita hacer una estimación realista del retardo instantáneo por enlace para aplicarlo en el mismo.

El modelo diseñado se basa en un análisis clásico de costes por enlace (costes variables con la carga) y la aplicación de un algoritmo de camino más corto (tipo Dijkstra) para seleccionar el mejor (el de menor coste, o lo que es lo mismo, el de menor latencia acumulada).

Por tanto, debemos trasladar las distintas componentes de retardo analizadas a una fórmula que asigne un coste equivalente a cada enlace. Sea $C_i(t)$, el coste asociado (latencia esperada) a la transmisión de una trama por el enlace L_i. Por analogía con la expresión (5.4), el coste asociado a un camino P_i en un instante de tiempo t, se puede calcular como:

$$C(P_i) = \sum C_j, \ \forall L_j \in P_i \qquad (5.9)$$

De nuevo, por analogía con las expresiones (5.5) y (5.6):

$$C_j = F(T_p + t_r + X_p + E[w]) = F_1(T_p + t_r) + F_2(E[t]) \qquad (5.10)$$

La expresión (5.10) representa el coste como el resultado de aplicar dos funciones diferentes:

F_1, a la parte del retardo independiente de la información transmitida. Se trata de un valor fijo para el enlace, que de acuerdo a las conclusiones previas puede ser irrelevante en muchos casos.

F_2, a la parte del retardo dependiente de la información transmitida (el tamaño medio de trama que determina el tiempo de transmisión y el tiempo de espera en cola que depende de la ocupación media del enlace). Se trata de un valor variable con la carga que aúna en un mismo dato los dos factores básicos que determinan la diferencia de latencia (retardo) en un camino: la velocidad de las líneas y su ocupación.

Por lo tanto, podemos reescribir la expresión (5.9) como:

$$C_j = K_j + F(E[t]) \qquad (5.11)$$

Expresión que como se puede deducir de los valores indicados en las tablas se puede simplificar en la mayoría de los centros de datos (salvo muy alta velocidad, ~ 100 Gbps), redes campus o metro

(salvo enlaces de mucha longitud, > 1 Km) y redes de interconexión (salvo enlaces de gran longitud como pueden ser los cables oceánicos) a:

$$C_j \sim F(E[t]) \qquad (5.12)$$

Para diseñar la función de costes *F*, dependiente de *E[t]*, debemos tener en cuenta dos factores: por un lado, la velocidad de la línea que determina el tiempo de transmisión (parte fija o valor de *E[t]* en un enlace vacío) y, por otro lado, el efecto de espera en cola derivado de la carga del enlace.

Vamos a considerar enlaces cuyo régimen binario *R* puede ir desde los 100 Mbps hasta 10 Gbps (velocidades típicas de redes Ethernet que aumentan en factores de 10). El IEEE propone en [8021D04] utilizar métricas de coste para la construcción de árboles de expansión con una dependencia inversa de *R*. Así, asignaremos un coste 1 al enlace de máximo *R* (de entre los valores considerados) y estado de carga vacío. Dicho coste aumentará en factores de 10 según disminuye el régimen del enlace.

Si utilizamos una referencia de 10 Gbps para asignar el coste unitario, podemos calcular el resto de costes para valores de R diferentes como:

$$Coste\ (R) = R_{10Gbps}/R = 10000/R \qquad (5.13)$$

Tabla 68. Valores de coste del modelo para enlaces de distintas velocidades binarias

R (Mbps)	Coste (sin carga)
100	100
1000	10
10000	1

Para añadir ahora el efecto de la carga, siguiendo la argumentación de la evolución del retardo de espera con respecto al de transmisión, el coste debería sufrir el mismo comportamiento exponencial con la carga (se duplica para valores de carga del 50%, etc.) Por tanto, podemos formular la expresión como:

$$Coste\ (R, \rho) = (10000/R) \bullet 1/(1-\rho) \qquad (5.14)$$

Y sustituyendo en (4.7) y (4.4) tenemos:

$$C_j = (10000/R_j) \bullet 1/(1-\rho_j) \qquad (5.15)$$

$$C(P_i) = \sum C_j = \sum (10000/R_j) \bullet 1/(1-\rho_j) \quad \forall L_j \in P_i \qquad (5.16)$$

Este será el modelo de costes básico para la representación de la latencia por enlace en la red. Lo denominaremos modelo exponencial puro o modelo *ExpHard* (EH).

5.4.2.1 Otros modelos de coste

Dado que el modelo M/M/1 de retardo representa el peor escenario de retardo posible, se plantea también incorporar a modo de contrapeso variaciones del mismo que reduzcan la influencia

de la carga de los enlaces en la latencia de los mismos y nos permitan comprobar la influencia real del modelo en los resultados obtenidos y caracterizar nuestro en condiciones menos agresivas de retardo esperado siempre en comparación con la cota que proporciona el modelo exponencial puro (EH).

De entre las muchas distribuciones posibles se ha optado por complementar el modelo de costes EH con dos nuevos modelos cuyas características de crecimiento de retardo frente a la carga sean mucho más suaves: los modelos exponencial suave y lineal.

El modelo exponencial suave (*ExpSoft*, ES), conserva la característica de crecimiento exponencial del modelo EH, pero con un efecto de la carga muy limitado que limita el crecimiento del coste a un máximo de dos veces el coste básico del enlace.

La expresión de cálculo utilizada es:

$$C_j = (10000/R_j)/(1-\rho_j/2) \qquad (5.17)$$

Que proporciona valores de 1,33 veces el coste de referencia del enlace para cargas del 50% y duplica el mismo para cargas del 99% o mayores.

En un punto intermedio entre ambos modelos exponenciales situamos el modelo lineal que contempla un crecimiento del coste lineal con la carga con una pendiente objetivo máximo de 10 veces el coste de referencia del enlace, es decir, un enlace completamente cargado tendría el mismo coste que uno completamente descargado cuya tasa binaria sea un orden de magnitud menor.

$$C_j = (10000/R_j) \bullet (1+9\bullet\rho_j) \qquad (5.18)$$

Como puede verse en la figura siguiente (eje de costes semi-logarítmico), este modelo presenta valores de coste superiores al modelo EH para cargas de menos del 90% y, sin embargo, no presenta el crecimiento tan grande del mismo de ahí en adelante.

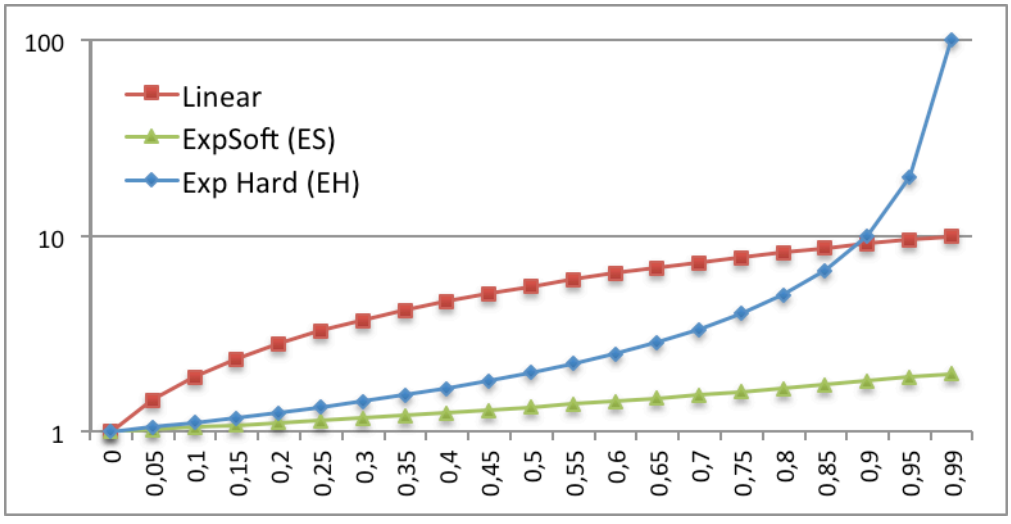


Ilustración 73. Características de crecimiento del coste asignado al enlace en función de la carga del mismo

5.4.3 Aleatorización de los costes

Los modelos presentados en la sección anterior proporcionan una estimación del coste a partir del valor de ocupación de los enlaces en el momento concreto en que se realice el cálculo. Para ello se ha recurrido a modelos matemáticos de teoría de colas que proporcionan un valor medio, pero que no reflejan un comportamiento instantáneo concreto. Por tanto, que el valor de ocupación medio de una cola (y por tanto el tiempo de espera resultante) tenga un valor determinado no implica necesariamente que la espera que sufra una trama concreta se corresponde directamente con ese valor medio.

Para representar este comportamiento, el propio modelo de costes incluye la posibilidad de aleatorizar el valor de coste que producen los modelos mencionados en la sección anterior tomando como referencia ese valor medio. Por ejemplo, con una distribución uniforme en un intervalo [valorMínimo..2•valor medio].

En resumen, el simulador permite elegir entre uno de los tres modelos de coste comentados tanto en modo fijo como en modo aleatorizado, dando lugar a seis modelos de coste diferentes que habrá que contrastar con distintas topologías y tipos de redes.

5.4.4 El modelo de flujos

Sirve para caracterizar la tasa de transferencia y la duración del mismo (a partir del volumen total de datos que transporte). Existen en la literatura distintos estudios que permiten caracterizar de una forma sencilla la naturaleza de los flujos de datos según el tipo de red (función y entorno organizativo) de que se trate. Los más destacables son :

- Tráfico POP (red de interconexión de ISP)
- Tráfico en centros de datos (DC)

5.4.4.1 Modelo de flujo POP (red de interconexión de ISPs)

Los parámetros que permiten caracterizar este tipo de flujos se han obtenido de [PD08] y [KDM09]. En estos artículos se construye un modelo de flujo a partir del análisis de trazas reales de las que se extraen las siguientes características:

Tasa de transferencia: define tres valores de velocidad binaria para los flujos y les asigna una cierta probabilidad cada uno:

- RF1 = 0,5 Mbps (60% de los flujos)
- RF2 = 1 Mbps (30% de los flujos)
- RF3 = 10 Mbps (10% de los flujos)

Volumen total de datos del flujo: cada flujo transporta una cantidad de datos variable que se caracteriza mediante una distribución Pareto con factor de forma (α=1,3) y valor inicial de 8 MB truncada en un valor máximo de 8 GB. El volumen medio de datos por flujo resulta de 34,8 MB.

5.4.4.2 Modelos de flujos de centros de datos

Los grandes centros de datos constituyen un escenario reciente y aún no muy bien conocido por razones de privacidad y seguridad. Los estudios recientes muestran una importante variación de la matriz de tráfico en el espacio y en el tiempo [KSG+09][BAA+09][GJK+09] diversidad derivada de las aplicaciones. Coexisten los flujos muy pequeños (RPCs de tipo transaccional) con flujos muy grandes (como *backups* o trabajos MapReduce [MR09]). Los centros de datos acaparan actualmente la atención dado que sus requisitos y escala son muy superiores a los habituales en las redes campus.

5.4.5 Modelo de tráfico

El modelo de tráfico permite caracterizar cómo se distribuyen por la red los orígenes y destinos finales de los distintos flujos de tráfico. Mediante una matriz de tráfico (TM) de *NxN* celdas, se asigna un peso relativo ω_{IJ} a cada par de nodos *I* y *J,* que representa la probabilidad de un flujo cualquiera de la red tenga al nodo *I* como origen y al nodo *J* como destino.

La matriz TM puede considerarse un parámetro de entrada del simulador (típico de configuraciones muy específicas como por ejemplo que solo se quiera considerar un determinado par <origen, destino> para todos los flujos), o calcularse a partir de un modelo de representación del peso de cada nodo como generador/receptor de tráfico en la red, los considerados modelos de gravedad.

En un modelo de gravedad, un nodo con una gravedad alta se convierte en un fuerte atractor de tráfico (fuente o destino). La gravedad de un nodo *I* (g_I) puede asignarse a partir de ciertos valores exógenos a la red (como es la cantidad de población que se concentra en el área de servicio de dicho nodo), o responder a un criterio de configuración de la red (por ejemplo en el caso de un servidor web o una gran base de datos en una red local), o simplemente seguir un modelo uniforme (todos los nodos son equivalentes) o puramente aleatorio (se reparten los pesos siguiendo algún criterio de asignación concreto).

A partir de los valores de gravedad de cada nodo se calcula la matriz TM simplemente multiplicando los pesos correspondientes a cada par origen destino:

$$\omega_{IJ} = g_I \cdot g_J \qquad (5.19)$$

En las distintas combinaciones de topologías y tipos de tráfico de las que se aportan resultados en la siguientes secciones se hará uso tanto de matrices TM pre-calculadas como de modelos de gravedad equiprobables, sesgados hacia uno o varios nodos de la red o sesgados según algún criterio de distribución concreto aplicable a la topología y configuración que se estudie en ese caso.

5.4.6 Estructura básica del simulador

El simulador está construido a partir de un único proceso software generador de flujos para toda la red. Independientemente del número de nodos y posibles pares <origen, destino>, hay una única entidad encargada de producir todo los flujos de datos que se van a simular en la red para, a continuación, asignarles el resto de características que los definen:

- Par de nodos origen y destino.

- Tasa de transferencia.
- Cantidad de datos que transporta (junto a la tasa de transferencia define la duración del flujo).
- Ruta seleccionada.

El proceso generador se modela con un tiempo entre nacimientos de flujo consecutivos (*IAT*) exponencial (proceso de Poisson generador de flujos), lo que dado el elevado número de flujos a simular (en el rango de cientos de miles para cada ejecución) no le resta generalidad.

Una vez hay un nuevo flujo en la red, se le asigna el par de nodos origen y destino atendiendo al modelo de distribución de flujos (modelo de tráfico) que corresponda en cada red simulada. La distribución de flujos en la red se modela mediante una matriz de tráfico (TM) que se calcula a partir del peso relativo de cada nodo de la red como atractor/fuente de datos siguiendo un esquema denominado modelo de gravedad del tráfico.

A continuación se le asigna una tasa de transferencia según el modelo de flujo que se disponga (normalmente una simple tabla de valores y probabilidades) y un volumen de datos total a transportar entre el origen y el destino, el tamaño del flujo, también predeterminado por el modelo de flujo en uso.

Finalmente, se selecciona una ruta entre los nodos origen y destino, la que presente una menor latencia en el momento de nacimiento del nuevo flujo. Mediante el modelo de costes que representa la latencia en cada enlace se asigna un valor a cada enlace de la red (válido para ese instante de tiempo, puesto que depende de la carga actual) y se aplica el algoritmo de Dijkstra para obtener el mejor camino posible (menor coste y, por tanto, menor latencia).

5.4.7 Topologías evaluadas

Se describen a continuación las topologías evaluadas y los resultados obtenidos. Se ha optado por utilizar variantes más o menos simplificados de la topología de la Ilustración 74. Se trata de una topología relativamente simple, un núcleo formado por una malla 2x2 de conmutadores centrales a los que se conectan desde cada lado cinco conmutadores de acceso (que sirven a los usuarios finales). Los conmutadores de acceso tienen enlaces a cada uno de los nodos centrales de su lado de la topología por lo que, en conjunto, tenemos una gran multiplicidad de caminos de coste equivalente para los flujos de datos que deban atravesar dicha topología. Este tipo de topología deriva de la arquitectura clásica de centros de datos con un plano de acceso, otro de distribución y otro final de servicio (casi una imagen especular del primero).

Posteriormente, vistos los positivos resultados en cuanto a la selección de caminos en función de la carga instantánea que íbamos obteniendo, se optó por analizar una nueva topología consistente en una malla plana 3x3 en la que la disponibilidad de caminos de igual coste es aún más acusada por lo que nos permite evaluar con mejor perspectiva las capacidades de diversidad de caminos (adaptación de carga) de ARP-Path.

Los enlaces centrales de la topología se modelan como canales dúplex de 1Gbps, sin retardo de propagación (son casi inapreciables en este entorno). Sin embargo, los enlaces de acceso tienen una velocidad de 10Gbps, lo que permite realizar excursiones de carga en el tramo central (desde muy baja a casi saturado) incluso con un solo conmutador de acceso activo (como es el caso en alguna de las simulaciones realizadas).

Todas las simulaciones tienen una duración de 128.000 segundos lo que garantiza (vistos los resultados) una validez estadística suficiente (con varianzas pequeñas). Cada grupo de resultados presenta resultados para distintos valores del tiempo entre llegadas (IAT) de flujo, parámetro clave del simulador que determina la cantidad de tráfico a tratar (solo hay un único generador de flujos). En cada caso, IAT toma distintos valores calculados para presentar al menos tres niveles de carga en la zona de la red que estemos estudiando (baja, media y alta).

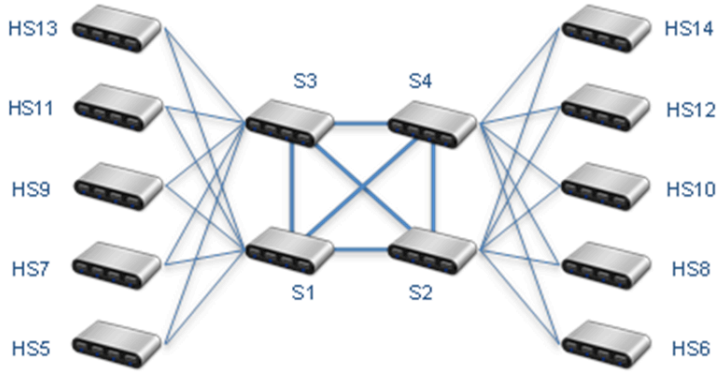


Ilustración 74. Topología clásica de un centro de datos

5.4.8 Resultados obtenidos

Los resultados que se presentan a continuación corresponden a variantes de la topología básica anterior. Primero se presenta un análisis comparativo de los distintos modelos de coste propuestos y se justifica la utilización del modelo EH (*Exponential Hard*) en los estudios posteriores. Mediante limitaciones en las secciones de acceso y servicio de la topología, se focalizan las siguientes simulaciones en el estudio de la capacidad de adaptación del protocolo ARP-Path a la carga instantánea de la red.

Por último, el tercer bloque de resultados corresponden a simulaciones sobre una topología en malla 3x3 con el objetivo concreto de caracterizar la capacidad de adaptación de carga del protocolo.

5.4.8.1 Estudio comparativo de los modelos de coste

Este primer grupo de resultados tienen por objeto mostrar las diferencias obtenidas al aplicar los modelos de coste definidos previamente: lineal, exponencial suave y exponencial abrupto. Además se incluye una iteración más incluyendo la posibilidad de aleatorizar cada modelo tal y como se ha detallado en la sección 5.4.3.

Las gráficas presentan solo la ocupación de los enlaces de la sección central (núcleo de cuatro conmutadores) en ambos sentidos, ya que se ha considerado que los flujos pueden originarse en cualquiera de las dos secciones laterales (acceso y servicio) de la topología.

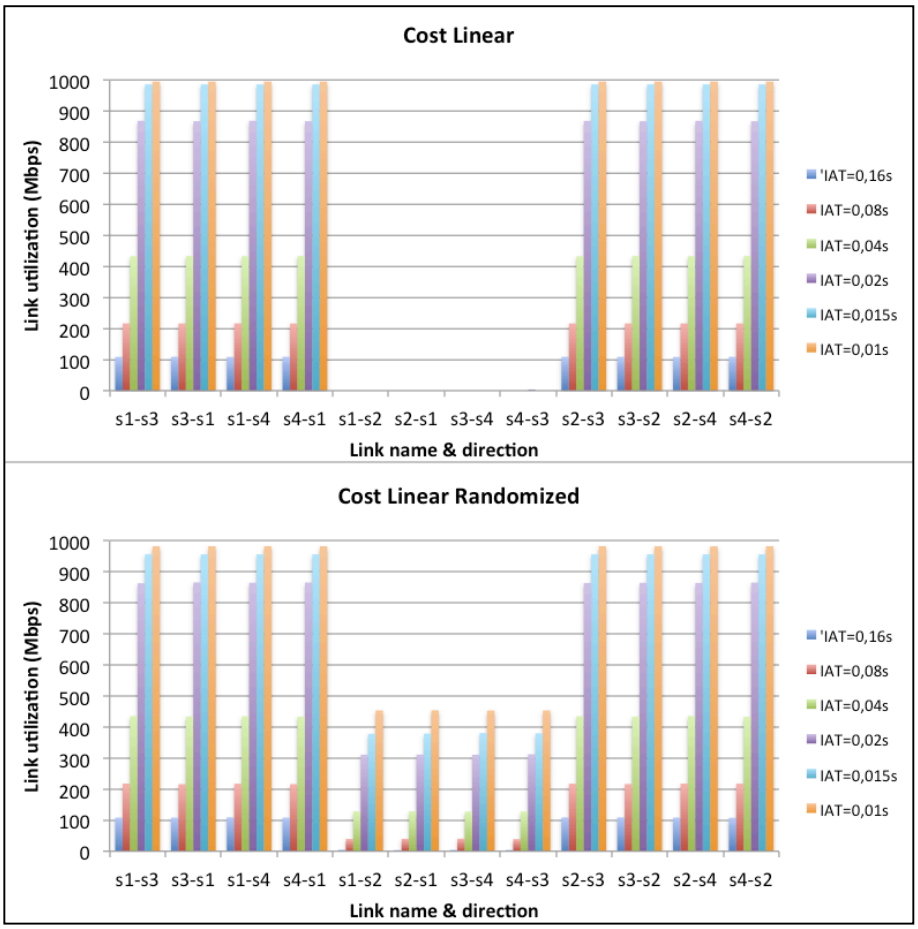


Ilustración 75. Comparativa de los modelos de coste Lineal (arriba) y lineal aleatorizado (abajo)

La Ilustración 75 muestra los resultados obtenidos con el modelo lineal puro y su variante aleatorizada. Puede observarse como en el caso del modelo lineal puro, todo el tráfico se cursa por los cuatro enlaces que cruzan el núcleo quedando completamente vacíos los enlaces laterales entre los dos nodos. Esto es debido a que el aumento de coste que implica utilizar un enlace lateral (añadir un segundo enlace al camino) no se compensa por el aumento de coste asociado a la ocupación de los enlaces de cruce directo del núcleo (el modelo lineal tiene una variación del coste con la utilización pequeño).

Sin embargo, cuando se utiliza el modelo aleatorizado, se producen dos efectos que, combinados, compensan la penalización en coste de utilizar el enlace lateral. Por un lado, se rompe la relación directa entre coste y ocupación del enlace (un aumento en la ocupación no produce necesariamente un aumento en el coste, al menos de forma instantánea, aunque si lo hará en valores medios). Por otro lado, se aumenta el rango de variación (incremento) del coste en relación a la ocupación por lo que, de nuevo, de forma instantánea se puede producir una penalización más fuerte a los enlaces de cruce directo que compense el coste de utilizar un enlace lateral (añadido).

La ilustración 76 muestra los resultados obtenidos para los modelos de coste exponencial suave (*ExpSoft*) y abrupto (*Expard*), ambos en versión aleatorizada. En este caso, la aleatorización no aporta suficiente variación en la función de coste como para que en el modelo suave compense la utilización de los enlaces laterales, mientras que el caso abrupto tiene un comportamiento aun más acusado que el lineal. De hecho, se produce un efecto de rebote o vuelta atrás resultado de combinar el propio crecimiento no lineal del modelo a cargas altas y el efecto de la aleatorización que termina resultando irrelevante.

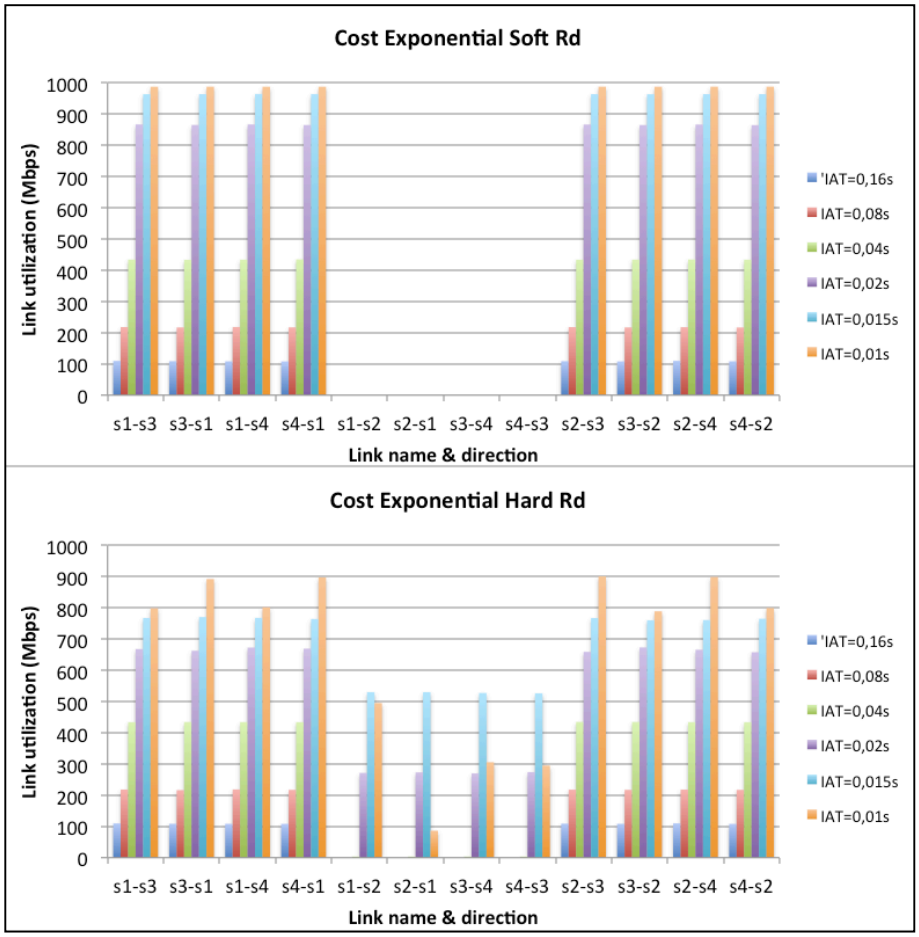


Ilustración 76. Comparativa de los modelos de coste aleatorizados exponencial suave y abrupto

Como conclusión general, podemos afirmar que el modelo exponencial abrupto nos permite capturar razonablemente el efecto de aumento no lineal del retardo de espera en cola de los puertos de salida de los conmutadores en relación a la utilización de los enlaces, sin tener que recurrir a la aleatorización de costes, más difícil de justificar físicamente.

5.4.8.2 Carga por enlace y distribución de rutas

Sobre la misma topología del apartado anterior, pero ahora simplificada a solo dos conmutadores, uno de acceso (h0) y otro de servicio (h1).

La Ilustración 77 muestra la distribución de carga en la red con modelo de costes lineal para valores de IAT de 0,64 0,32 0,16 0,08 y 0,04 s. Solo se muestran los resultados del modelo lineal a modo de ejemplo. La gráfica incluye la carga de los enlaces de acceso y servicio a ambos extremos de los enlaces núcleo. Como era de esperar la ocupación de estos enlaces (única vía de salida/llegada de todos los flujos) es bastante más alta que los del núcleo central en donde el tráfico se reparte entre varios caminos. Puede observarse un reparto casi perfecto entre los cuatro enlaces que permiten atravesar el núcleo de lado a lado. De la misma forma, en los enlaces de acceso desde cada conmutador de host el tráfico también se reparte entre las dos vías disponibles equitativamente.

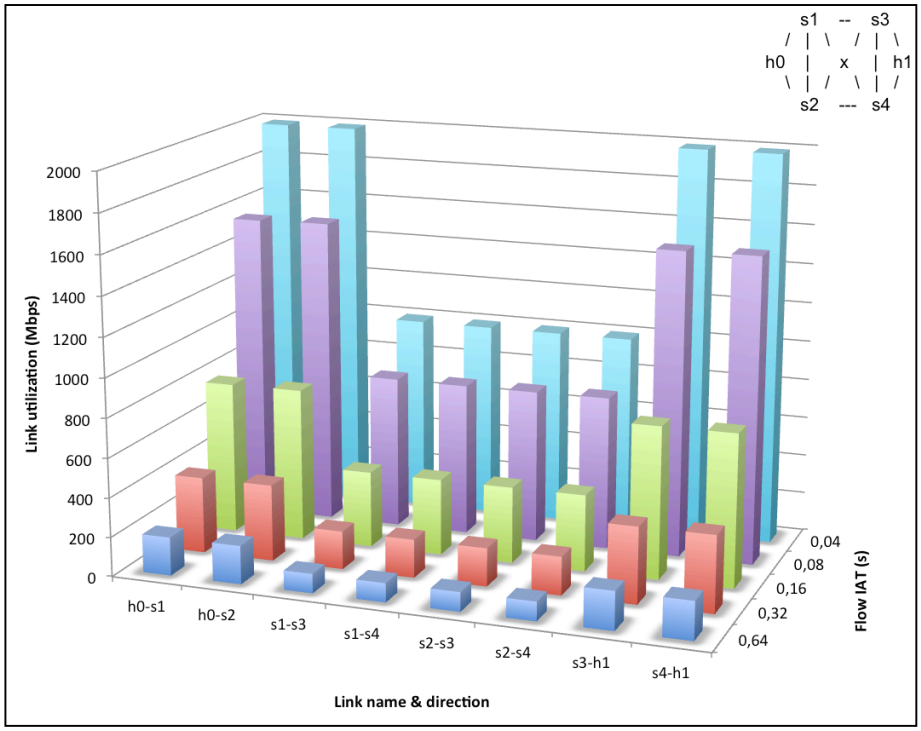


Ilustración 77. Utilización de los enlaces con caminos alternativos de distinto coste

Para estudiar el reparto de carga cuando los caminos alternativos no son de igual coste se utiliza la topología anterior pero los conmutadores de host h0 y h1 tienen solo un enlace, a s1 y s3 respectivamente. De esta manera, los caminos alternativos incurrirán en uno o dos saltos más dando lugar a rutas de uno, dos y tres saltos. En este caso, se ha optado por mostrar los resultados obtenidos con el modelo exponencial abrupto que consideramos más representativo.

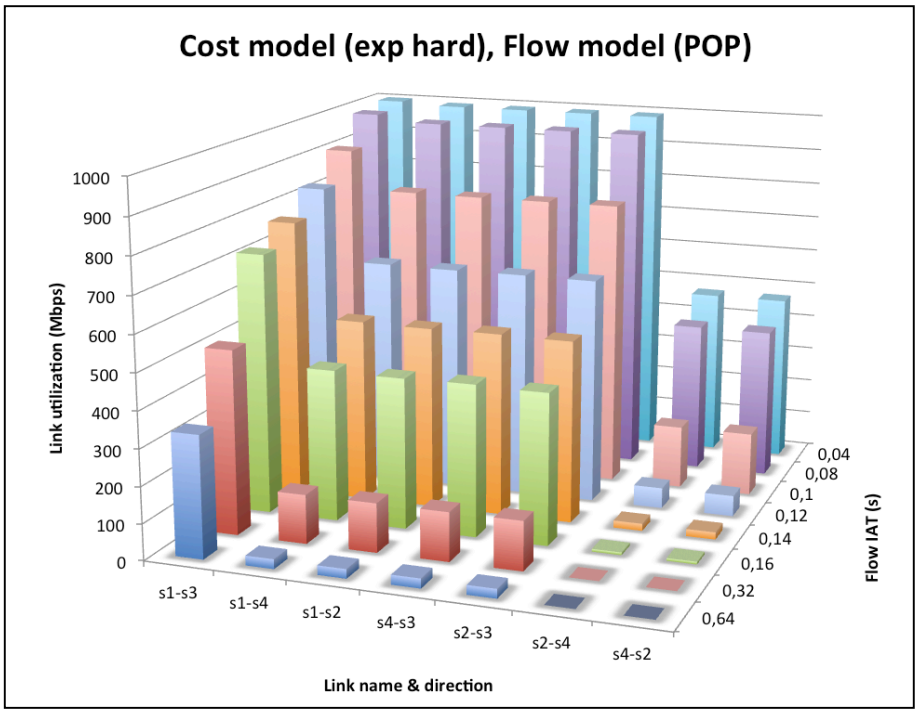


Ilustración 78. Utilización de los enlaces de acceso y núcleo en topología simplificada

La Ilustración 78 muestra los resultados en cuanto a ocupación de los enlaces. Se observa como la ruta directa (a través del enlace S1-S3) es el camino predilecto, ya que solo realiza un salto para atravesar el núcleo de lado a lado.

Conforme aumenta la carga, los caminos alternativos (S1-S4-S3 y S1-S2-S3), con dos saltos, comienzan a utilizarse cargando los enlaces respectivos. Para cargas muy altas, incluso la ruta (S1-S2-S4-S3), con tres saltos, comienza a utilizarse.

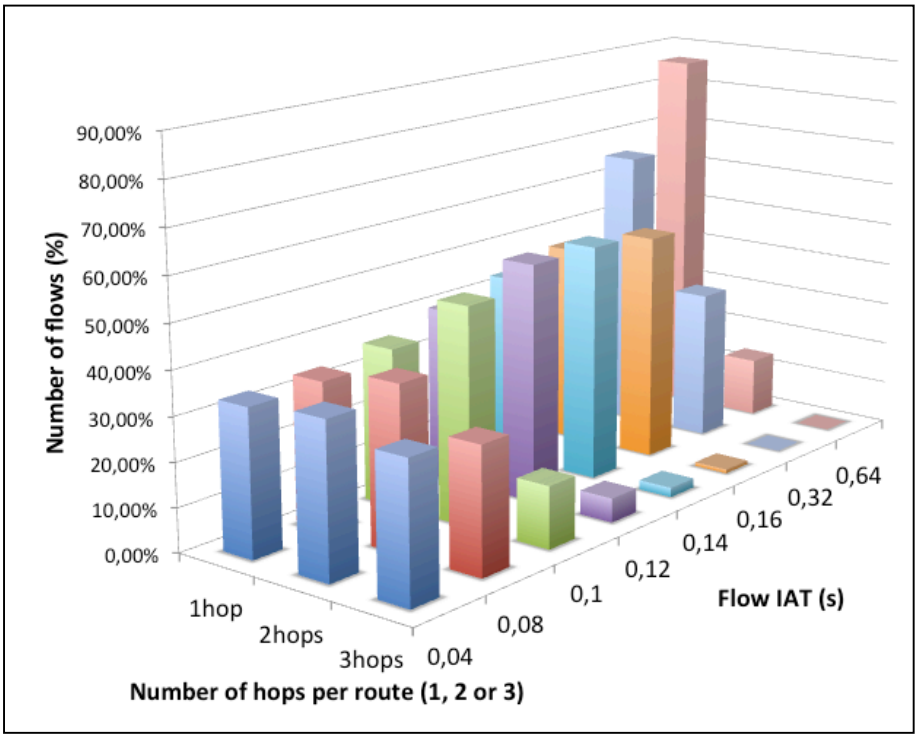


Ilustración 79. Porcentaje de flujos que utilizan rutas de uno, dos y tres saltos

La Ilustración 79 muestra los mismos resultados pero representando el porcentaje de flujos que atraviesan el núcleo en rutas de uno, dos y tres saltos. Aunque puede parecer sorprendente que el porcentaje de flujos a dos saltos supere el de los directos, cabe recordar que existen dos caminos diferentes con dos saltos (S1-S2-S3 y S1-S4-S3) disjuntos por uno solo (S1-S3) con un solo salto.

5.4.8.3 Estudio de la distribución de carga

Los resultados obtenidos hasta el momento indican una capacidad de adaptación del protocolo a la carga de la red muy importante. Con el objeto de estudiar este comportamiento más en detalle se optó por utilizar una nueva topología, que se prestase a facilitar este análisis. Se eligió una malla plana de tres por tres conmutadores. En función de dónde se sitúen las fuentes y los destinos del tráfico que se simule, esta topología ofrece múltiples alternativas (caminos) todas ellas con el mismo número de saltos (o igual coste). Todos los enlaces tienen la misma capacidad (1Gbps, salvo indicación expresa concreta).

Se incluyen en las simulaciones matrices de distribución de tráfico uniformes (flujos aleatorios de cualquier origen a cualquier destino), extremo a extremo (*edge-to-edge*, E2E), con flujos entre los nodos 0 y 8 exclusivamente, y lado a lado (*side-to-side*, S2S), con flujos entre los nodos 3 y 5 exclusivamente.

Se utiliza el modelo POP de flujo [PD08] y una variante del mismo consistente en flujos de uno o dos órdenes de magnitud superior (en tasa) con el objetivo de aportar una granularidad más apreciable en la ocupación relativa de un flujo sobre un enlace y, por tanto, en su influencia sobre el cálculo de costes. Se utilizan el modelo de costes exponencial abrupto (EH) simple.

La Ilustración 80 muestra los resultados con tráfico uniforme. Apenas hay variaciones de utilización entre enlaces.

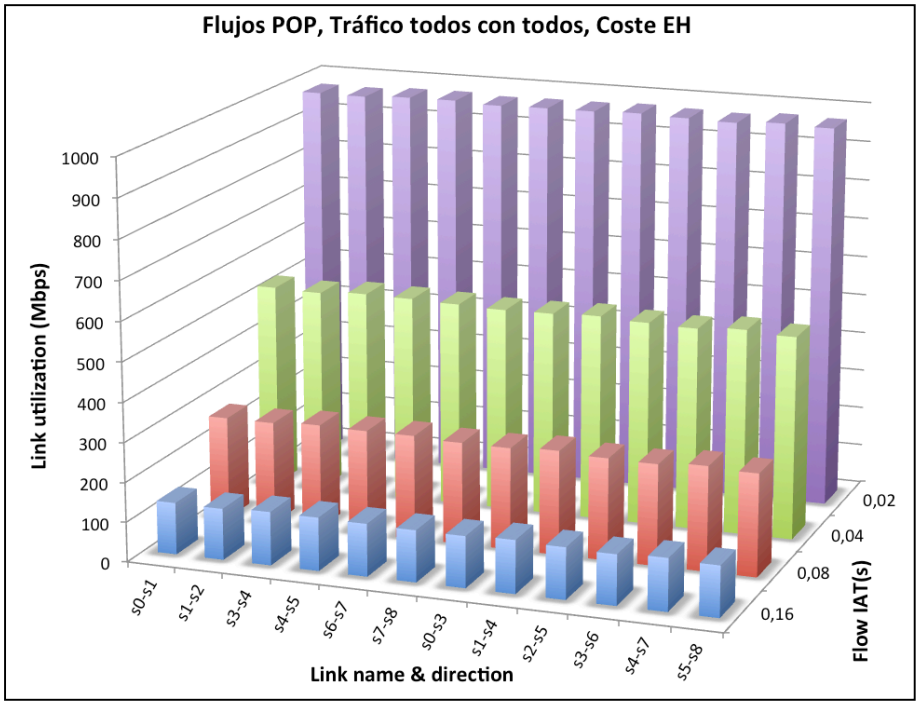


Ilustración 80. Tráfico uniforme (todos con todos)

La Ilustración 81 muestra los resultados obtenidos con el modelo aleatorizado. Se puede observar que apenas hay variaciones, si acaso un leve aumento en la utilización de los enlaces centrales de la malla que corresponden con caminos más cortos.

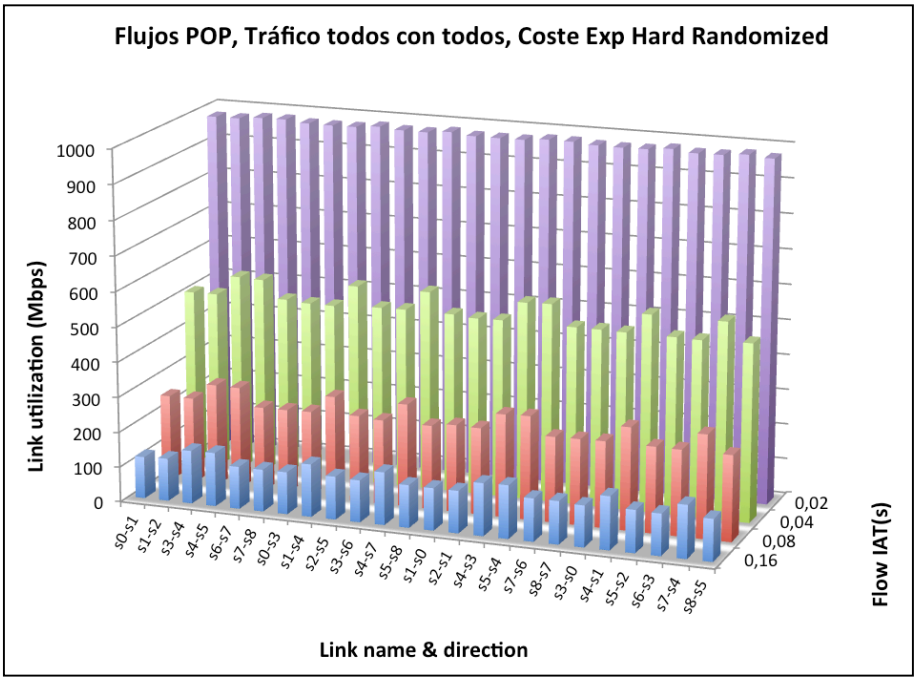


Ilustración 81. Tráfico uniforme (todos con todos) con coste aleatorizado

Las gráficas siguientes muestran los resultados obtenidos con tráfico localizado entre dos nodos extremo a extremo, entre S0 y S8 (tráfico E2E) y lado a lado, entre S3 y S5 (tráfico S2S).

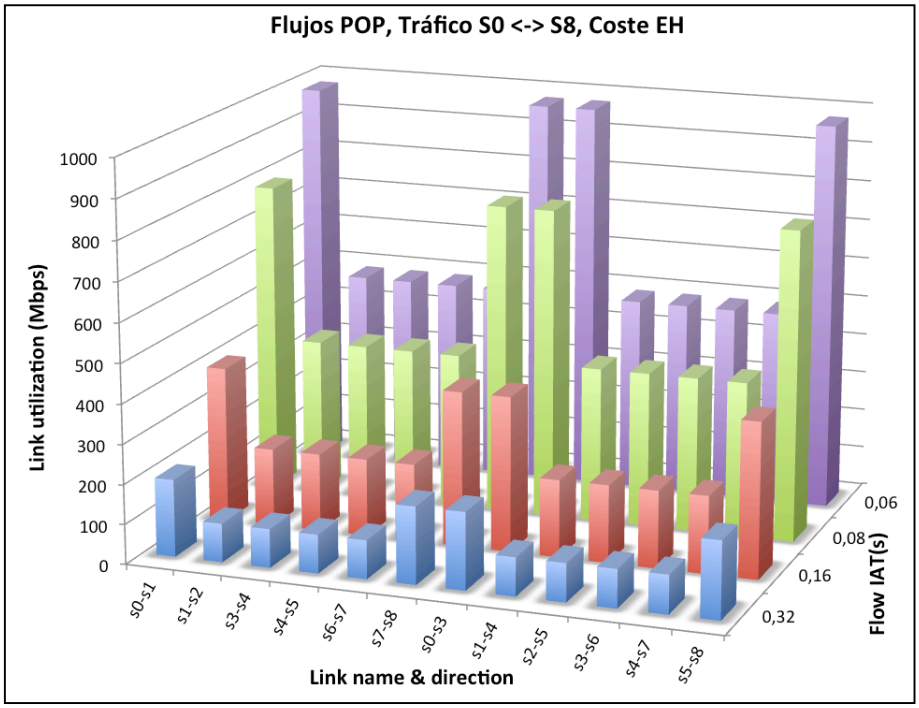


Ilustración 82. Distribución de tráfico con flujos E2E

En el modelo E2E de la Ilustración 82 todas las rutas disponibles atraviesan 4 enlaces, por tanto, tienen un coste inicial equivalente. Los nodos S0 y S8 solo tienen dos enlaces, por tanto, acumulan el doble de tráfico que los demás ya que todos los flujos deben atravesar uno de los dos. La distribución de carga entre enlaces homogéneos es equilibrada.

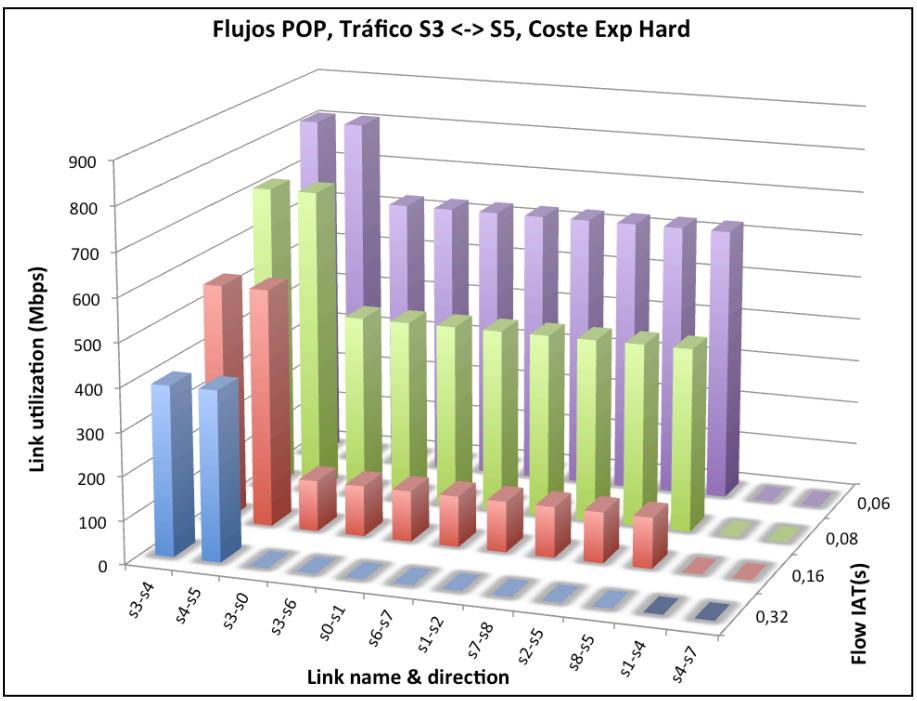


Ilustración 83. Distribución de tráfico con flujos S2S

En el modelo lado a lado, el sentido S3->S5 dispone de una ruta de 2 saltos: S3-S4-S5 y dos de 4 saltos: S3-S0-S1-S2-S5 y S3-S6-S7-S8-S5, todas ellas disjuntas entre sí. La Ilustración 83 muestra el reparto de carga hacia las rutas de 4 saltos conforme aumenta la carga de la ruta preferente. Los enlaces S1<->S4 y S4<->S7 no se usan.

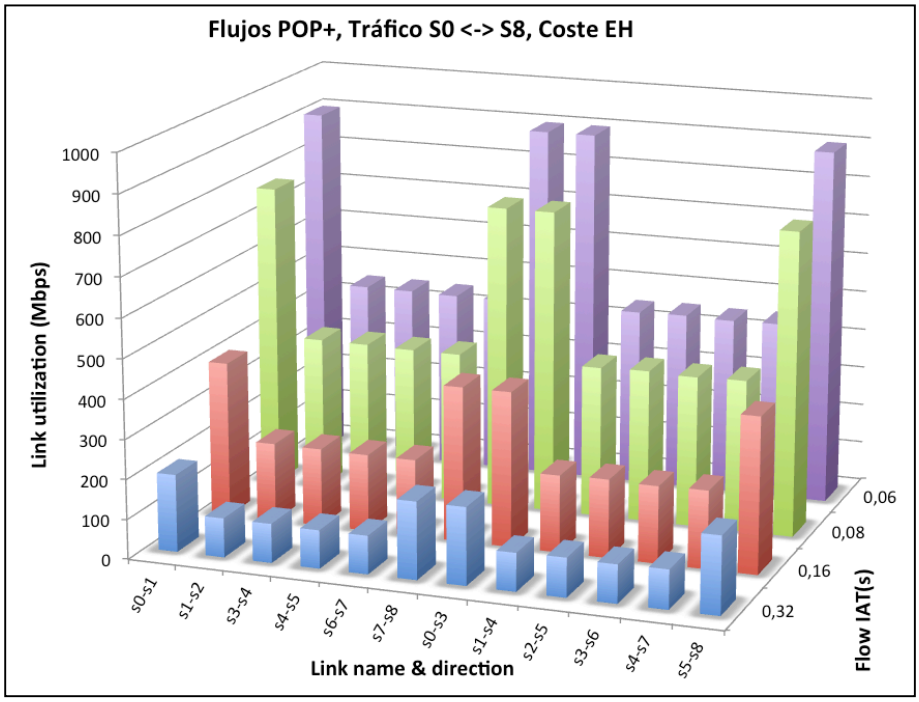


Ilustración 84. Tráfico extremo a extremo con modelo de flujos POP+

También se han realizado simulaciones aumentando en uno y dos órdenes de magnitud la tasa binaria de los flujos unitarios del modelo POP, con el objetivo de ver el efecto sobre la carga de los enlaces (y, por tanto, sobre el modelo de costes) de manejar flujos más grandes. En este caso, la adición de un nuevo flujo (o su eliminación) de una ruta tiene una influencia en la utilización de los enlaces de la ruta mucho mayor.

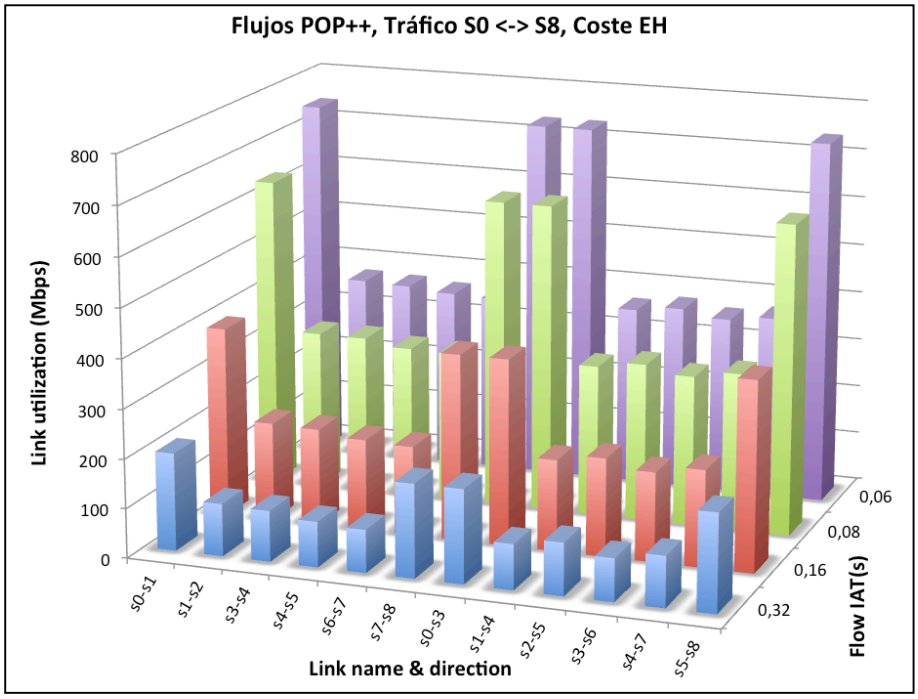
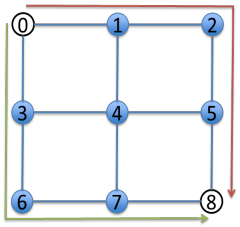


Ilustración 85. Tráfico extremo a extremo con modelo de flujos POP++

Sin embargo, los resultados de la Ilustración 84 apenas muestran diferencias con respecto al modelo POP, más allá de un ligero aumento en la varianza de los resultados por lo que solo se incluyen gráficos representativos del caso extremo a extremo (E2E).

Y para el modelo POP++, aumentando de nuevo el régimen binario de los flujos en otro orden de magnitud, tenemos los resultados de la Ilustración 85.

El siguiente grupo de simulaciones utiliza caminos pre-calculados, cuyo coste se asigna en función de la capacidad disponible del enlace más cargado del camino (para verificar que la selección de camino se hace por coste de enlace y no por caminos completos). Se han realizado tres grupos de simulaciones con dos, cuatro y seis caminos precalculados disponibles para tráfico E2E (como en la Ilustración 82). La Ilustración 86 muestra el reparto perfecto de los flujos entre los dos caminos posibles.



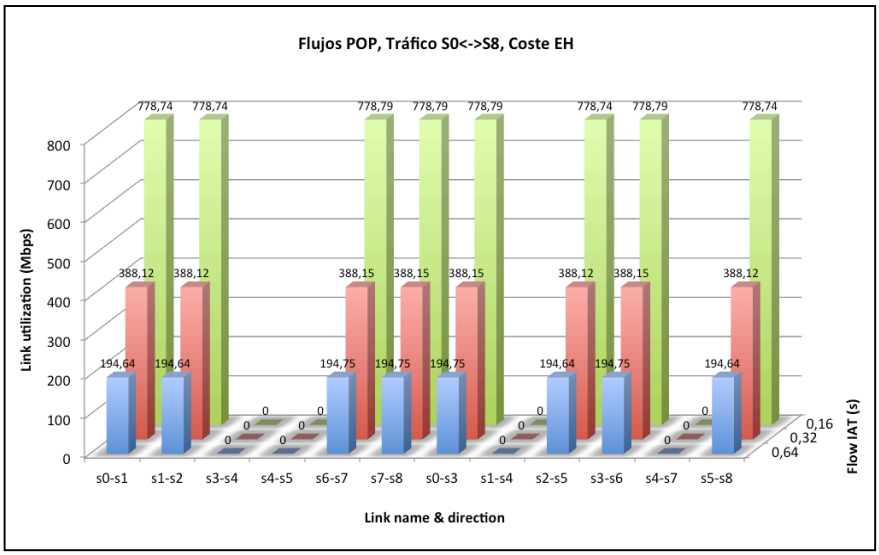
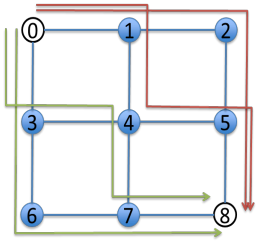


Ilustración 86. Distribución de tráfico con cuatro caminos precalculados (esquema de la izqda.)

Cuando se dispone de cuatro caminos, como en la Ilustración 87, se observa de nuevo un reparto equitativo entre los dos enlaces de salida (desde 0) y de llegada (hasta 8) y en los dobles caminos intermedios para cada rama.



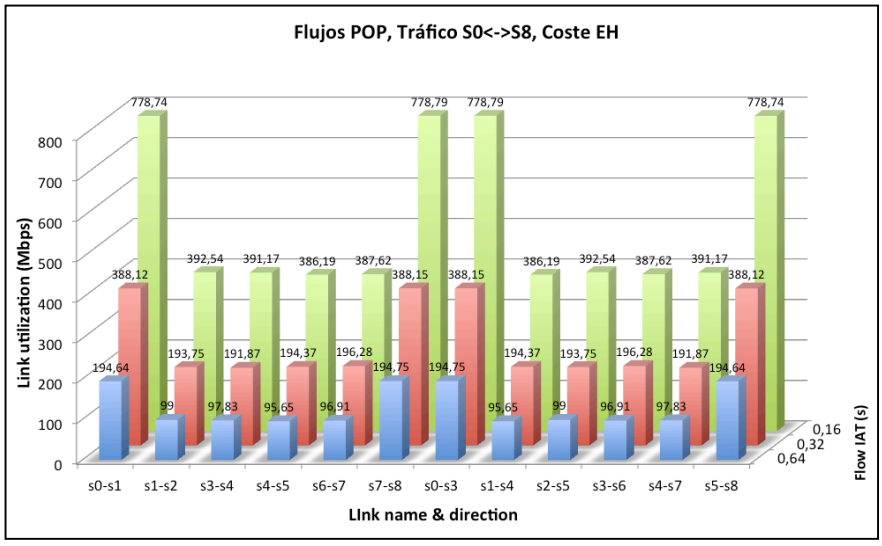


Ilustración 87. Distribución de tráfico con dos caminos precalculados (esquema de la izqda.)

Y, por ultimo, para el caso de seis caminos, la interacción entre rutas se complica ya que aparecen caminos que cruzan de un lado a otro la red, tanto en sentido vertical como en horizontal. Debido a la propia estructura de la malla, tres de los seis caminos posibles se concentran en cada uno de los dos enlaces de salida desde el nodo S0 y de los dos enlaces de llegada al nodo S8. Por tanto, el cuello de botella será siempre uno de esos cuatro enlaces decantando la elección de ruta hacia uno de los otros tres caminos. Entre ellos, el desempate lo marcará uno de los dos enlaces posibles del nodo opuesto al cuello de botella anterior, provocando una distribución de tráfico con una característica de proporción uno a tres entre los enlaces de los nodos opuestos al tráfico (S2 y S6) frente a los enlaces centrales (nodo S4) de la topología. De ahí las ostensibles diferencias entre la Ilustración 82 y la Ilustración 88.

El último grupo de simulaciones se centra en el modelo de tráfico uniforme, se establecen flujos desde hosts conectados a cualquiera de los conmutadores, con destino aleatorio (todos con todos). Paulatinamente se irán introduciendo variaciones en la distribución mediante el modelo de gravedad (para observar el comportamiento de los flujos según vayamos concentrando el tráfico global de la malla en zonas concretas de la misma).

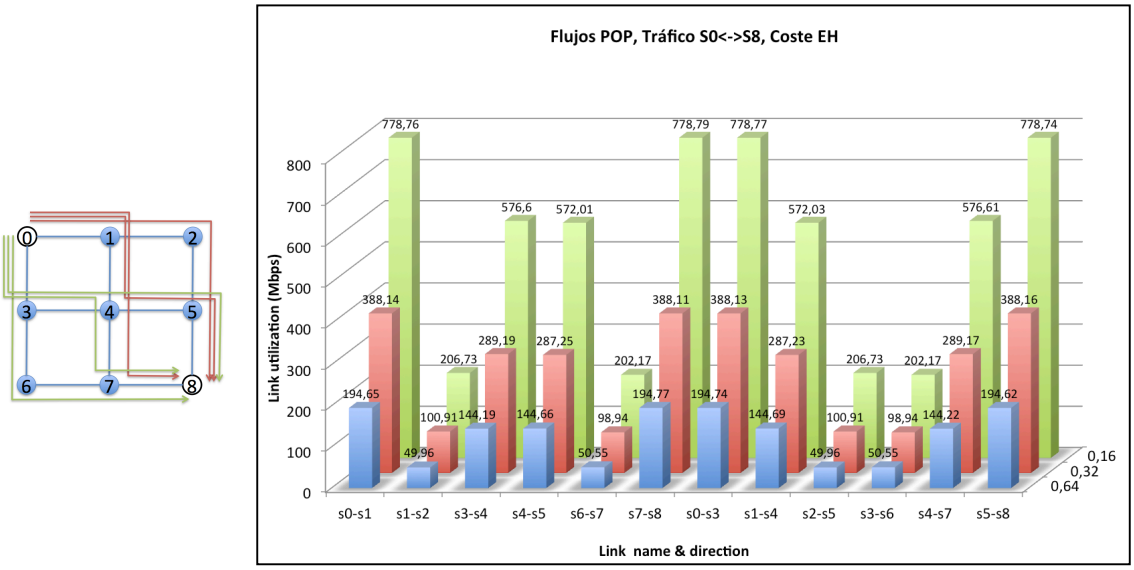


Ilustración 88. Distribución de tráfico con seis caminos precalculados (esquema de la izqda)

Sin embargo, pese a que el tráfico es uniforme, nos vamos a fijar exclusivamente en el estudio de los flujos originados en el conmutador S0 y cuyo destino sea un host conectado al conmutador S8 (modelo E2E). Sobre el gráfico que representa la propia topología de la malla se representan, asociados a cada enlace tres valores representativos: la ocupación del enlace en cada sentido (en Mbps) y el porcentaje de flujos de S0->S8 que atraviesa dicho enlace. Las ilustraciones muestran los resultados obtenidos para el nivel de carga más bajo de los simulados (IAT=0,16s), mientras que las tablas recogen las ocupaciones obtenidas aumentando la carga, para distintos valores de IAT.

En conjunto, se muestran los resultados obtenidos en tres casos concretos: con tráfico uniforme, con tráfico sesgado hacia el nodo S6 (abajo-izquierda) y con tráfico sesgado hacia los nodos S6 y S2 (extremos opuestos al par origen destino de los flujos S0-S8 que vamos a rastrear).

La Ilustración 89 muestra los resultados obtenidos para el caso uniforme. Se observa cómo la ocupación de los enlaces refleja una distribución de carga equitativa en toda la red, independientemente de si se trata de un enlace central o de uno de los periféricos de la topología. Si nos fijamos en los flujos de S0 a S8, la figura indica (en verde) el porcentaje de flujos que pasa por cada enlace.

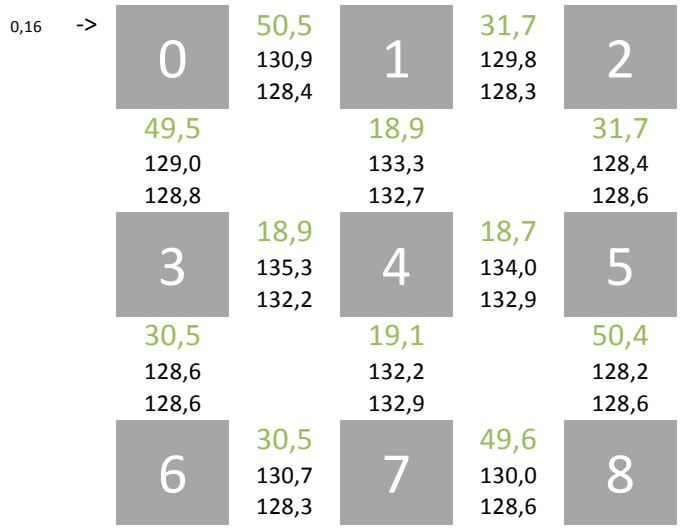


Ilustración 89. Distribución de flujos S0->S8 y ocupación global de enlaces (IAT=0,16s), con pesos uniformes

Podemos ver como los enlaces de salida y llegada reflejan se reparten por igual los flujos mientras que el resto de enlaces presenta una distribución sesgada hacia los enlaces periféricos, esto es debido a que los cuatro enlaces centrales concentran necesariamente todos los flujos con origen/destino el nodo S4, por lo que, aumentando su ocupación y coste, y forzando el resto de flujos por otros caminos que no atraviesen el centro.

La Tabla 69 muestra la ocupación de los enlaces de la malla para distintos valores de carga global (IAT desde 0,16 a 0,024 s). No se observa una variación apreciable con la carga global.

Tabla 69. Tráfico por enlace para distintos valores de carga global con pesos uniformes

	Utilización del enlace (Mbps)											
IAT (s)	**s0-s1**	**s1-s2**	**s3-s4**	**s4-s5**	**s6-s7**	**s7-s8**	**s0-s3**	**s1-s4**	**s2-s5**	**s3-s6**	**s4-s7**	**s5-s8**
0,16	130,9	128,95	129,76	133,3	135,3	128,56	128,38	133,99	132,23	130,65	128,22	130,02
0,08	259,97	259,25	259,06	264,14	264,85	257,41	258,92	264,02	263,08	260,18	258,37	259,36
0,04	520,04	518,81	517,69	524,17	525,26	516,63	518,14	523,58	523,44	520,36	517,36	517,43
0,024	866,11	866,54	865,72	873,7	873,41	864,27	866,58	873,48	873,06	866,61	865,17	866,22

En la siguiente figura se muestran los valores obtenidos con un modelo de gravedad que desbalancea el tráfico global de la red hacia el nodo S6 (esquina inferior izquierda de la malla). Se asigna un factor de gravedad $g_{S6}=4$, mientras que el resto mantienen un factor unitario. De esta manera, a la hora de asignar un par de nodos <origen, destino> a cada nuevo flujo generado en la simulación, el nodo S6 tendrá cuatro veces más probabilidad de ser elegido como origen o destino que los demás.

La Ilustración 90 muestra cómo el protocolo trata de equilibrar la carga de los enlaces evitando en la medida de los posible los enlaces cercanos a S6, sin embargo, no es posible hacer un equilibrio total, ya que S6 es origen y destino de cuatro veces más flujos que el resto de nodos. El resultado final es un cierto gradiente de crecimiento de los valores de ocupación de los enlaces en función de su cercanía a S6. Si nos centramos en lo que ocurre con los flujos de S0 a S8, vemos como son

literalmente repelidos hacia enlaces lo más alejados posible de S6, hasta el punto que la ruta más alejada (a través del nodo S2) concentra casi el 70% de este tráfico.

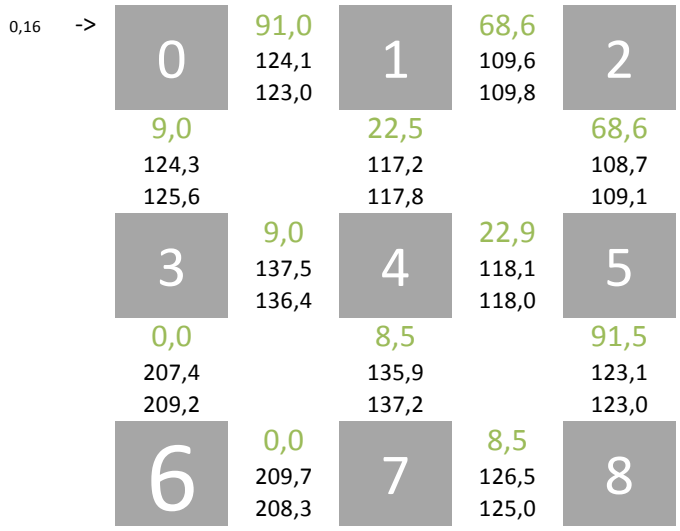


Ilustración 90. Distribución de flujos S0->S8 y ocupación global de enlaces (IAT=0,16s), con pesos sesgados hacia S6

La Tabla 70 muestra la variación en porcentaje de flujos S0->S8 que atraviesa cada enlace de la red (a diferencia de la tabla anterior que mostraba la ocupación de cada enlace) en función de la carga global. Podemos apreciar como el efecto de repulsión de los flujos hacia el nodo S2 se va incrementando paulatinamente según aumenta la carga, pues el coste de los enlaces cercanos a S6 crece más rápidamente que el de los demás.

Tabla 70. Tráfico por enlace para distintos valores de carga global y pesos sesgados hacia S6

IAT (s)	Flujos de S0 a S8 en el enlace (%)											
	s0-s1	**s1-s2**	**s3-s4**	**s4-s5**	**s6-s7**	**s7-s8**	**s0-s3**	**s1-s4**	**s2-s5**	**s3-s6**	**s4-s7**	**s5-s8**
0,16	91,0	68,6	9,0	22,9	0,0	8,5	9,0	22,5	68,6	0,0	8,5	91,5
0,08	94,0	74,0	6,0	19,9	0,0	6,1	6,0	20,1	74,0	0,0	6,1	93,9
0,04	95,8	77,6	4,2	18,2	0,0	4,2	4,2	18,2	77,6	0,0	4,2	95,8

Por último, en este tercer caso, el tráfico se concentra en los nodos S2 y S6, ambos con un peso de cuatro, de manera que los dos extremos contrarios de la red al origen y destino de los flujos en estudio, se convierten en atractores de tráfico sobrecargando los enlaces que les rodean (en relación a los demás).

Al tener dos focos de tráfico situados en perpendicular al camino directo entre S0 y S8, los flujos de este camino se concentran en la zona central de la red, evitando ambos extremos opuestos (ningún flujo atraviesa S2 ni S6). La distribución de carga es equitativa en los enlaces que no tienen acceso directo a S2 y S6 y, como en el caso anterior, es menor que en estos debido a que los flujos entrantes y salientes de estos dos nodos necesariamente deben atravesarlos (el protocolo no puede desviarlos por otro sitio).

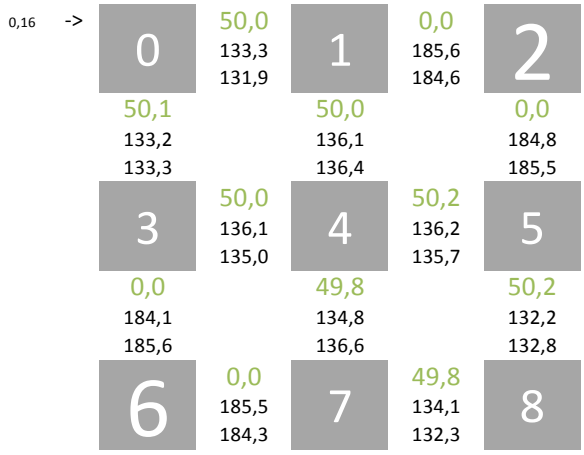


Ilustración 91. Distribución de flujos S0->S8 y ocupación global de enlaces (IAT=0,16s), con pesos sesgados hacia S6 y S2

En la Tabla 71 podemos ver como el aumento de carga tampoco produce en este caso efectos apreciables sobre la distribución de tráfico.

Tabla 71. Tráfico por enlace para distintos valores de carga y pesos sesgados hacia S6 y S2

	Flujos de S0 a S8 en el enlace (%)											
IAT(s)	**s0-s1**	**s1-s2**	**s3-s4**	**s4-s5**	**s6-s7**	**s7-s8**	**s0-s3**	**s1-s4**	**s2-s5**	**s3-s6**	**s4-s7**	**s5-s8**
0,16	50,0	0,0	50,0	50,2	0,0	49,8	50,1	50,0	0,0	0,0	49,8	50,2
0,08	49,9	0,0	50,1	49,7	0,0	50,3	50,1	49,9	0,0	0,0	50,3	49,7
0,04	50,2	0,0	49,8	50,2	0,0	49,8	49,8	50,2	0,0	0,0	49,8	50,2

5.4.8.4 Estudio de la distribución de carga (simulador de paquetes)

Con el fin de validar los resultados del modelo conceptual de simulación mediante flujos, se ha construido un simulador de paquetes, implementado en OMNeT++. Sobre este nuevo simulador, aprovechando las librerías de generación de tráfico TCP/IP del sistema, se han intentado replicar los estudios mencionados en las secciones anteriores.

Se han realizado simulaciones sobre la malla de 3x3 nodos con enlaces a 100Mbps (en lugar de 1Gbps) y para valores de IAT de 1,6 segundos (proporcionalmente, la ocupación del sistema debe ser la misma ya que se reduce tanto la velocidad de los enlaces como la tasa de generación de flujos. Aún así, al modelar como paquete del flujo individualmente, la carga de proceso de cada ejecución es tan exagerada que solo podemos realizar simulaciones limitadas a un máximo de 20.000 segundos de tiempo simulado (y con tiempos de simulación superiores a la semana).

La Ilustración 92 muestra los resultados obtenidos con tráfico exclusivo entre los nodos S0 y S8 (son tráfico de fondo) en cuanto a porcentaje de flujos por cada enlace (izquierda) y ocupación de los enlaces (en sentido S0 a S8). Podemos apreciar cómo el reparto no es tan perfecto y presenta una mayor variabilidad entre ejecuciones debido a que los tiempos de simulación no permiten alcanzar una precisión y confianza en los resultados similar a la del simulador de flujos.

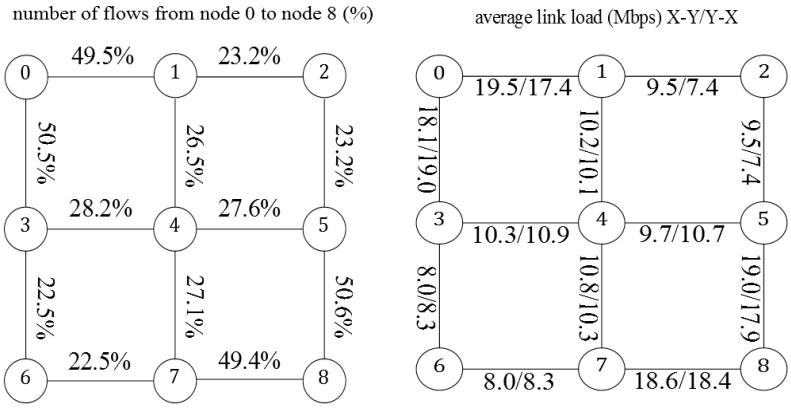


Ilustración 92. Distribución de carga y flujos S0->S8 sin tráfico de fondo

En las siguientes figuras (Ilustración 93 e Ilustración 94) se presentan los resultados obtenidos con tráfico de fondo uniforme (izquierda) y sesgado hacia el nodo S6, con el mismo factor 4 del simulador de flujos (derecha). Podemos ver cómo la ocupación de los enlaces se reparte apreciablemente aunque se destaca una carga ligeramente mayor en los enlaces centrales.

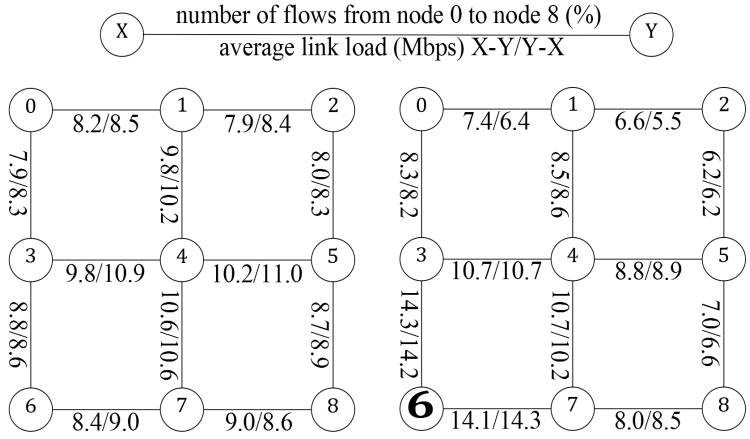


Ilustración 93. Distribución de carga con tráfico de fondo uniforme (izqda.) y sesgado hacia S6 (dcha.)

Respecto de los flujos de S0 a S8, se aprecian también los efectos de reparto y repulsión mostrados con el simulador de flujos aunque en grado mucho menor.

La evaluación combinada, mediante los simuladores de flujos y de paquetes, de la capacidad de adaptación del encaminamiento a la distribución de carga instantánea de la red está aceptada para su publicación en [ICR+13].

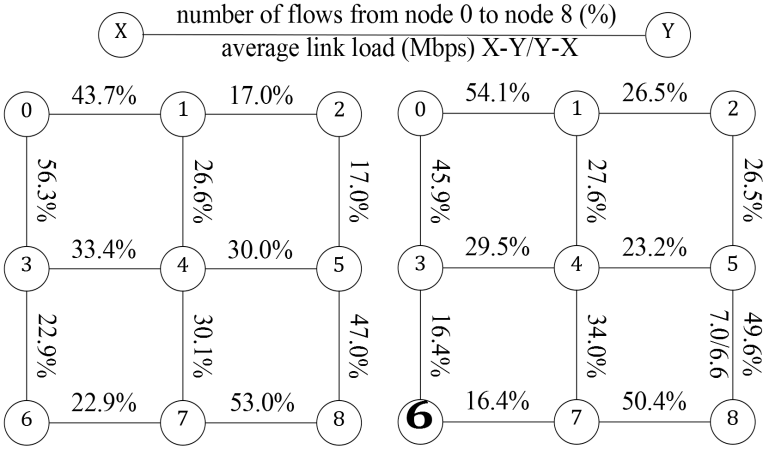


Ilustración 94. Distribución de flujos S0->S8 con tráfico de fondo uniforme (izqda.) y sesgado hacia S6 (dcha.)

5.5 Conclusiones

ARP-Path es una evolución del paradigma de puentes transparentes que permite la obtención de caminos mínimos sin bloquear enlaces ni usar protocolos de encaminamiento convencionales como estado de enlaces o vectores distancia. Los puentes ARP-Path utilizan tramas de difusión Ethernet estándar para encontrar los caminos más rápidos entre hosts. Las evaluaciones muestran un rendimiento muy superior al del árbol de expansión y similar al de enrutamiento por camino más corto en cuanto a retardos y mejor en términos de rendimiento y complejidad del protocolo. Se ha implementado una prueba de concepto en Linux. Las evaluaciones iniciales muestran un potencial muy interesante para la distribución de la carga de tráfico que se investigará más adelante. La combinación del protocolo ARP-Path con la función Etherproxy en los puentes frontera puede mejorar la escalabilidad del protocolo mediante la reducción radical de la difusión de tramas.

El generador de flujos propuesto hace posible la simulación de complejas matrices de tráfico configurables cuya evaluación con simuladores de paquetes es casi inviable por el tiempo y procesado requeridos, varios órdenes de magnitud mayor. El generador es asimismo susceptible de extensiones de gran interés para aumentar la precisión. Las simulaciones intensivas en curso con el simulador de paquetes permitirán determinar con precisión las diferencias de resultados esperables entre ambos simuladores y validar la adecuación de las futuras mejoras del simulador de flujos.

6 Conclusiones y trabajos futuros

En este capítulo final se recogen a modo de resumen ejecutivo las principales aportaciones resultado del trabajo realizado en esta Tesis Doctoral y detalladas en los capítulos 3, 4 y 5: el protocolo HURP (encaminamiento mediante prohibición de giros y direccionamiento jerárquico), los protocolos de la familia TRE (TRE, TRE+ y DTRE) y el protocolo ARP-Path.

También se incluyen algunas propuestas de trabajo futuro para continuar con la investigación aquí expuesta y en los que ya nos encontramos trabajando dentro del grupo de investigación GIST-Netserv.

6.1 Encaminamiento mediante prohibición de giros y direccionamiento jerárquico

HURP es un protocolo de conmutadores autoconfigurable y escalable basado en el protocolo RSTP, el cual es extendido para la asignación de direcciones jerárquicas a los puentes y que utiliza el protocolo *Up/Down (U/D)* para evitar bucles de tramas. La complejidad computacional del protocolo es baja $O(N*d^2)$, mucho menor que otras propuestas que requieren conocimiento global de la topología. Puede implementarse como arquitectura única o de forma combinada pero independiente, coexistiendo con puentes estándar 802.1D en el mismo dispositivo mediante el direccionamiento local jerárquico HLMAC separado de las direcciones globales MAC. Además, permite el reenvío rápido de tramas por el árbol de expansión empleando el direccionamiento jerárquico mediante descodificación paso a paso de la dirección destino, sin necesidad de tablas de encaminamiento.

El mecanismo complementario de encaminamiento por atajos es simple y admite diversas métricas. El mecanismo de prevención de bucles de HURP es aplicable a redes Ethernet de cualquier velocidad y a redes de interconexión. Las prestaciones son similares a otros protocolos como TP y TBTP con mucha menor complejidad y consistentemente mejores que *U/D*, más cercana a conmutadores de camino más corto (*shortest path*) y no depende de manera significativa de la elección del puente raíz, aunque son posibles heurísticos de optimización con precálculo y elección previa del puente raíz.

La influencia relativa del número de giros prohibidos en las prestaciones disminuye al aumentar el grado medio de la red porque el número de giros prohibidos por nodo es $d*(d-1)/2$ siendo d el grado medio del nodo, aumentando cuadráticamente con d por lo que al aumentar el grado de algunos nodos se crean muchos caminos alternativos en una topología. Por ello, la prohibición de algunos giros afecta en menor medida a la longitud de los caminos.

Como resumen del estudio del porcentaje de giros prohibidos podemos concluir que el tipo de topología tiene una gran influencia en el valor nominal de todos los protocolos estudiados en la comparación (HURP, U/D y SP) pero no produce un cambio relativo en el rendimiento comparado entre ellos. Tanto U/D como HURP presentan una dependencia con la raíz elegida limitada (valores máximo y mínimo) y siempre dentro de límites razonables. Al utilizar U/D es frecuente preseleccionar como candidatos adecuados a ser raíz para limitar este efecto. En HURP no parece necesario ya que incluso en redes sin escala (con gran dispersión de grados) los nodos de grado alto se colocan naturalmente en posiciones cercanas a la raíz independientemente de cuál sea la raíz elegida. HURP proporciona resultados consistentemente mejores que U/D para redes de grado medio entre 4 y 8 y tiende a los mismos valores para grados mayores (redes de grado medio 8 o más resultan poco prácticas y difíciles de encontrar en la realidad).

Con respecto al estudio comparativo de rendimiento, cabe destacar que en las redes de topología aleatoria sin escala, HURP proporciona rendimientos cercanos a SP en todos los casos estudiados, superando además los resultados de U/D desde un 10% nominal para redes pequeñas hasta casi un 20% en las de mayor tamaño. Es destacable que no hay variaciones apreciables con el grado medio de la red. Nuevamente, la explicación tiene que ver con el hecho de que los nodos de mayor grado son "empujados" hacia arriba en el árbol por estar muy conectados, independientemente de cual sea la raíz elegida, un efecto no estudiado en la literatura sobre U/D y que se investiga en el grupo.

Finalmente, en relación al tamaño de los caminos medios seleccionados, HURP muestra valores de camino medio consistentemente mejores que U/D y muy cercanos a SP, prácticamente idénticos

en el caso de las topologías sin escala. En cualquier caso, las variaciones entre HURP, U/D y SP con respecto a este parámetro no son grandes en ninguna de las topologías estudiadas.

6.2 Protocolos TRE, enrutamiento Ethernet basado en árbol

La arquitectura TRE (*Tree-based Routing Ethernet*), compuesta por los protocolos TRE, TRE+ y D-TRE, desarrollados sucesivamente se propone para solventar las ineficiencias más importantes de STP/RSTP, derivadas del bloqueo de todos los enlaces ajenos al propio árbol de expansión, lo que produce una utilización de la infraestructura muy baja, caminos medios largos (con alto retardo) y acumulación de rutas en las zonas altas del árbol (en las proximidades del nodo raíz), en general, serios problemas de escalabilidad. A la vez, se mantiene el objetivo de evitar la presencia de bucles, punto fuerte de RSTP y debilidad de los protocolos de tipo *shortest path* (estado de los enlaces) que requieren mecanismos adicionales.

Comparados con el estándar RSTP, los protocolos de la Arquitectura TRE ofrecen mayor rendimiento, computado como el número de flujos existente en los enlaces "cuello de botella". Comparados con protocolos de tipo *shortest path*, ofrecen mayor protección contra bucles a la vez que reducen la complejidad en su implementación. En este sentido, los protocolos TRE+ y D-TRE alcanzan valores de rendimiento muy cercanos a *shortest path*.

TRE es un protocolo basado en el árbol de expansión, que utiliza los atajos que constituyen los enlaces con nodos vecinos no pertenecientes al árbol. Los *conmutadores* TRE no necesitan tabla de reenvío para encaminar las tramas de datos ya que únicamente comparan la distancia al destino que proporcionan los vecinos con la que ofrece el árbol por defecto. Por lo tanto, el alcance del conocimiento topológico de los conmutadores TRE es de un salto. TRE consigue rutas más cortas que STP, el cual utiliza estrictamente el árbol, sin que se vea penalizado el coste del encaminamiento.

Con el protocolo TRE+ se incrementa el conocimiento topológico hasta los nodos situados a dos saltos, por lo que se descubre mayor cantidad de atajos. Para conseguirlo, cada conmutador TRE+ realiza un intercambio de mensajes entre sus vecinos, del tipo "vector distancias", y almacena la información topológica recibida en una tabla de reenvío, que consultará por cada trama recibida que sea necesario reenviar hasta el destino. El rendimiento de TRE+ es considerablemente mayor respecto a TRE, a costa de aumentar moderadamente la complejidad computacional en el reenvío de las tramas.

Por último, el protocolo D-TRE implementa un conocimiento topológico variable en función de características de la red (grado medio) y del propio nodo (grado). Implementar un alcance variable conlleva que los nodos deben intercambiar información entre ellos para descubrir otros nodos situados a más de un salto de distancia. En TRE+ se utiliza un mensaje "vector distancias" porque los nodos descubren siempre a nodos situados a dos saltos. Sin embargo, en D-TRE el objetivo es que este alcance pueda ser todavía mayor. Por lo tanto, es necesario el intercambio de mensajes del tipo "estado de enlaces".

Se han diseñado y evaluado dos variantes del protocolo, D-TRE1 y D-TRE2. En D-TRE1, se ha considerado que un nodo con un número alto de vecinos dispondrá de información suficiente de la red para realizar un encaminamiento eficiente, no necesitando ampliar su alcance a más de un salto de distancia. El umbral de número de vecinos (grado) en el que se considera que un nodo dispone de información suficiente se ha fijado en el grado medio de la red. En el protocolo D-TRE1, los nodos con grado mayor que la media de la red solo utilizarán la existencia de sus vecinos (el alcance es un salto), mientras que el resto descubrirán la red por cada una de sus ramas hasta que se encuentren con un nodo con grado mayor (el alcance se extiende hasta dos o tres saltos).

Por su parte, en el desarrollo de D-TRE2, se consideró que en los niveles superiores del árbol es donde se concentra la mayoría de los enlaces y, por lo tanto, de atajos, por lo que es fundamental ampliar el alcance en estos niveles. Los nodos intentarán utilizar los atajos existentes en los niveles superiores del árbol, con el fin de reducir el tamaño de las rutas, cuya consecuencia es la liberación de carga en los enlaces de la raíz, que suelen ser los más ocupados. Por lo tanto, el alcance de conocimiento de los nodos se va incrementando a medida que se asciende por el árbol.

La evaluación comparativa de ambos protocolos ha tenido en cuenta factores de sobrecarga, estabilidad, tamaño medio de las rutas seleccionadas y rendimiento. D-TRE1 tiene 2,5 veces menos sobrecarga que D-TRE2 y requiere menor proceso para el encaminamiento de las tramas (un 32% de consultas menos) en redes grandes (512-1024 nodos). DTRE-2 alcanza un mayor rendimiento (el doble de *Throughput relativo*) y rutas un 2% más cortas, además, necesita menor proceso que D-TRE1 (12% de consultas menor) en redes pequeñas (64-128 nodos).

6.3 Protocolo ARP-Path

ARP-Path aporta una evolución conceptual de los puentes transparentes con aprendizaje que consideramos importante en la evolución de los puentes ya que no requiere un protocolo auxiliar de enrutamiento (ni de árbol de expansión) en capa dos, a diferencia de las propuestas actualmente en estandarización (o recientemente estandarizadas) como RBridges y Shortest Path Bridges. En los conmutadores ARP-Path cada host establece un camino mínimo al mismo tiempo que se envía el paquete estándar *ARP-Request*, pero inundado por todos los enlaces. El camino marcado por el primer paquete *ARP-Request* que alcanza el destino se confirma aprovechando el paquete *ARP-Reply* de respuesta del host destino. ARP-Path mantiene la estructura de trama Ethernet y es compatible con puentes estándar en modo núcleo-isla.

Los conmutadores ARP-Path establecen caminos bajo demanda, lo que significa que las rutas obtenidas se adaptan a las condiciones de carga de la red de cada momento sin necesidad de utilizar ningún mecanismo de balanceo de carga. Por diseño, cuando se solicita un nuevo camino, el camino más rápido para llegar al host de destino será el seleccionado. Esto significa que los nuevos caminos seleccionados hacia hosts (distintos de los hosts activos existentes comunicantes) seguirán el camino más rápido en el momento de la solicitud, por lo tanto, las rutas con mayor carga (retardo) no serán seleccionadas si existen rutas alternativas con menor retardo. En su lugar, tenderán a ser seleccionados los enlaces con latencia más baja. Debido al elevado número de hosts, como el camino establecido tiene granularidad por cada host, el tráfico quedará equilibrado en toda la infraestructura, lo que resulta en latencias menores.

La capacidad de adaptación de caminos a la carga, nativa del protocolo ARP-Path, se ha estudiado en detalle mediante la implementación de un simulador de red basado en flujos de información entre pares de hosts (no modela la granularidad de paquete) con el objetivo de reducir la ingente cantidad de recursos de proceso y memoria necesarios en un simulador clásico de paquetes. El simulador incorpora modelos de coste que emulan el comportamiento de retardo sensible a carga clave para la elección de los caminos y modelos de distribución de flujos entre los elementos de red basados en matrices de tráfico con pesos gravitatorios.

La evaluación del protocolo se ha complementado con una implementación de conmutadores ARP-Path en Linux y en la plataforma *Openflow* [OFS] con tarjetas *NetFPGA [NetFPGA]* (actualmente en pruebas), así como simulaciones sobre Omnet++[OMNET] desarrollados por otros investigadores del grupo. Asimismo, estos desarrollos han servido para validar los resultados obtenidos mediante el simulador de flujos.

Las prestaciones obtenidas son similares o ligeramente superiores a las de enrutadores de camino mínimo y muy superiores a STP. Las pruebas con tráfico real y de reconfiguración muestran las robustez y rapidez de adaptación del protocolo así como sus cualidades como ecualizador de carga basado en retardo.

6.4 Trabajos futuros

Al igual que las contribuciones que conforman esta Tesis, el trabajo futuro que se plantea entronca directamente con la línea central de trabajo del grupo de investigación GIST-Netserv, que en este momento se focaliza en la evolución del concepto fundamental de ARP-Path y su aplicación al desarrollo de nuevos conmutadores de altas prestaciones.

En este trabajo, juega un papel muy importante el simulador de red basado en flujos que habrá que adaptar para la evaluación de otros aspectos importantes del protocolo como puede ser la reconfiguración y para incorporar nuevos modelos de tráfico que recojan algunas características novedosas de los flujos de información en entornos característicos como son los centros de datos. Los centros de datos, cada vez tienen más importancia y suponen por su peculiar arquitectura un campo de aplicación muy prometedor para protocolos como ARP-Path.

Adicionalmente, con respecto a los protocolos de la familia TRE, aunque no en la línea central de trabajo del grupo actualmente, quedaría por explorar la posibilidad de unificar las dos variantes del protocolo dinámico en un único protocolo que determine el conocimiento topológico que debe adquirir cada nodo de la red en función tanto de la relación entre su grado y el grado medio de la red, como de su situación en la estructura del árbol de expansión que se construya.

En cuanto a la evolución de los protocolos TRE, es de destacar, que en el marco de otra Tesis Doctoral del grupo, se ha desarrollado un protocolo para centros de datos (Torii-HLMAC) basado en el direccionamiento (múltiple en este caso) TRE, cuyas prestaciones se continuarán estudiando dado su interesante potencial (autoconfigurable, autorreparación de caminos con *track-back*, completamente distribuido).

Abreviaturas

10 GE. 10 Gigabit Ethernet.

ARP. Address Resolution Protocol.

ATM. Asynchronous Transfer Mode.

CST. Common Spanning Tree.

CIST Common and Internal Spanning Tree.

DHCP. Dynamic Host Configuration Protocol.

EIGRP. Enhanced Interior Gateway Routing Protocol

ES-IS. End System to Intermediate System (protocol).

FE. Fast Ethernet.

GARP. Generic Atribute Registration Protocol.

GE. Gigabit Ethernet.

GOE. Global Open Ethernet.

HTTP. Hypertext Transfer Protocol.

ICMP. Internet Control Message Protocol.

IDF. Intermediate Distribution Frame.

IEEE. Institute of Electrical and Electronic Engineers.

IETF. Internet Engineering Task Force.

IGMP. Internet Group Management Protocol.

IS-IS. Intermediate System to Intermediate System.

IST. Internal Spanning Tree.

LLDP. Link Layer Discovery Protocol.

MDF. Main Distribution Frame..

MIB. Management Information Base.

MPLS. Multiprotocol Label Switching.

MSCI. Multiple Spanning Tree Configuration Identifier**.**

MSTI. Multiple Spanning Tree Instance.

MSTP Multiple Spanning Tree Protocol.

NAT. Network Address Translator.

NDP. Neighbour Discovery Protocol.

PAE. Protocolo de Árbol de Expansión.

P2P. Peer- to-Peer.

RSTP. Rapid Spanning Tree Protocol IEEE 802.1D 2004.

SAN. Storage Area Network.

SEND. Secure Neighbour Discovery Protocol.

SDH. Synchronous Digital Hierarchy.

SPB. Shortest Path Bridging (IEEE).

STAR. Spanning Tree Alternate Routing Protocol

STP. Spanning Tree Protocol IEEE 802.1D 1998.

VDSL. Very high rate Digital Subscriber Line.

VLAN. Virtual LAN.

Bibliografía y Referencias

[8021ad] IEEE 802.1ad Provider Bridges. http://www.ieee802.org/1/pages/ 802.1ad.html

[8021ag] IEEE 802.1ag - Connectivity Fault Management, http://www.ieee802.org/ 1/pages/802.1ag.html, 2007.

[8021ah] IEEE 802.1ah Provider Backbone Bridges. http://www.ieee802.org/1/pages/802.1ah.html

[8021aq] IEEE 802.1aq™. Draft Standard for Local and Metropolitan Area Networks-- Virtual Bridged Local Area Networks-- Amendment 4: Shortest Path Bridges. IEEE, 2011.

[8021D04] IEEE 802.1D-2004 IEEE standard for local and metropolitan area networks, Media access control (MAC) Bridges, IEEE, 2004, Disponible en [Internet]: <http://standards. ieee.org/getieee802/802.1.html > (Julio 2005)

[8021D98] IEEE 802.1D.IEEE-1998 IEEE standard for local and metropolitan area networks-- Common specifications--Media access control (MAC) Bridges.

[8021Q03] IEEE 802.1Q-2003 IEEE standard for local and metropolitan area networks - Virtual Bridged Local Area Networks- IEEE, 2003. Disponible en [Internet]: <http://standards. ieee.org/getieee802/download/802.1Q-2003.pdf > (Julio 2005).

[8021X] IEEE 802.1X- IEEE standard for local and metropolitan area networks. Port based network access control. IEEE. Disponible en [Internet]: http://www.ieee802. org/1/ pages/802.1x.html.

[8023] 802.3™. IEEE Standard for Information technology— Telecommunications and information exchange between systems—Local and metropolitan area networks—

[AB02] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks", Rev. Mod. Phys. no. 74, pp. 47-97, 2002.

[BAA+09] Benson, T., Anand, A., Akella, A., and Zhang, M., "Understanding Data Center Traffic Characteristics", In Proceedings of ACM WREN, 2009.

[BA99] A.L. Barabási and R. Albert, "Emergence of Scaling in Random Networks", Science, pp. 509–512, Oct. 1999.

[Bar02] A.L. Barabási, "The New Science of Networks", Perseus, Cambridge, MA 2002.

[BCF+94] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawlk, C. L. S. ad J. N. Seizovic, and W. Su, “Myrinet: A Gigabit-per-second local area network”, IEEE Micro, vol. 15, pp. 29–36, Febrero 1994.

[BG92] D. Bertsekas & R. Gallagher, “Data Networks, 2nd Ed.”, Prentice-Hall, 1992.

[BRITE] Boston University Representative Internet Topology Generator. Disponible en [Internet]: http://www.cs.bu.edu/brite/

[Cas+94] S. Casale, V. Catania, A. Puliafito, L. Vita, “A Remote Bridging Technique to Increase Performability in Distributed Systems”, En IEEE Transactions on Industrial Electronics, Vol. 41, pp. 461-472, Abril 1994.

[CGMP] Cisco Group Management Protocol, Disponible en [Internet]: http://www. cisco.com/warp/ public/473/22.html (Julio 2005).

[CIG+10] Juan A. Carral, Guillermo Ibáñez, Alberto García-Martínez, Miguel A. López-Carmona, Iván Marsá-Maestre, "TRE+: Extended Tree-Based Routing Ethernet," ETRI Journal, vol.32, no.1, Feb. 2010.

[CLJ01] C.-S. Chang, D.-S. Lee, Y.-S. Jou, “Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering”, En IEEE HPSR '01, Dallas, Mayo 2001.

[Chi+02] G. Chirovolu et al., “Encapsulation schemes to extend Ethernet to Metropolitan Area Networks”, Alcatel Telecommunications Review, 3rd Quarter 2002.

[Chi+04] G. Chirovolu et al., “Issues and Approaches on Extending Ethernet Beyond LANs”, IEEE Communications Magazine, pp. 80-86, March 2004.

[Cia05] R. Ciampa, “Layer 3 switching basics”, Disponible en [Internet]: http://www. pulsewan.com/data101/pdfs/layer3_switching.pdf (Julio 2005).

[CIS05] Cisco Enterprise Marketing, “Hierarchical Campus Design”, Disponible en [Internet]: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns24/c643/cdccont0900/ aecd800d8129.pdf (Mayo 2005).

[CIS11] “Understanding Switch Latency”, White Paper June 01, 2011. Disponible en [internet]: http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps11541/white_paper_c11-661939.html

[CPV93] V. Catania, A. Puliafito, L. Vita, “A Modular Network Architecture for Performance Enhancement in Extended Local Area Networks”, IEEE Transactions on Reliability. Vol 42, no. 1, March 1993.

[CT97] H. Chi and C. Tang, “A deadlock-free routing scheme for interconnection networks with irregular topologies," in International Conference on Parallel and Distributed Systems, 1997.

[DM78] Dalal, S., Metcalfe, R., “Reverse path forwarding of broadcast packets”, Communications of the ACM Vol. 21, No. 12 pp. 1040-1048, December 1978.

[DP88]R. C. Dixon, D. A. Pitt, “Addressing, bridging, and source routing [LAN interconnection], IEEE Network, Volume: 2, Issue: 1, Jan 1988, pp. 25-32.

[Dro97] R. Droms, “Dynamic host configuration protocol. Internet Engineering Task Force”, Request for Comments: 2131, Marzo 1997.

[DS87] W.J. Dally, C.L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks", IEEE Transactions on Computers vol. C-36 no.5, pp187-196, May 1987.

[Dua91] J. Duato, "On the Design of Deadlock-Free Adaptive Routing Algorithms for Multicomputers: Design Methodologies," Proc. Parallel Architectures and Languages Europe 91, June 1991.

[DYN97] J. Duato, S. Yalmachili, L. M. Ni, "Interconnection networks: An Engineering Approach", Los Alamitos, California: IEEE Computer Society, 1997.

[EC09] Elmeleegy, Khaled and Cox, Alan L., "EtherProxy: Scaling The Ethernet By Suppressing Broadcast Traffic", Proceedings of IEEE INFOCOM 2009, Rio de Janeiro, Brazil.

[FE04] M. Fiddler G. Einho, "Routing in Turn-Prohibition Based Feed-Forward Networks", En NETWORKING 2004, Lecture Notes in Computer Science, LNCS 3042, p.1168 –1179, 2004.

[FFF99] C. Faloutsos, P. Faloutsos, M. Faloutsos, "On Power-Law Relationships of the Internet Topology", En Proceedings of the ACM SIGCOMM 1999, p. 251-262, Sept. 1999.

[Fin05] N. Finn, "Shortest Path Bridging", Julio 2005, Disponible en [Internet]: www. ieee802.org/802_tutorials/july05/nfinn-shortest-path-bridging.pdf

[Gar12] J. García del Río, "D-TRE: Protocolo dinámico de encaminamiento en Ethernet basado en árbol jerárquico de expansion con direccionamiento jerárquico", Proyecto Fin de Carrera dirigido por J. A. Carral, Universidad de Alcalá, 2012.

[GDS03] R. García, J. Duato, F. Silla, "LSOM: A Link State Protocol Over MAC Addresses for Metropolitan Backbones Using Optical Ethernet Switches", Proceedings Second IEEE NCA'03.

[GDS98] R. García, J. Duato, and J. Serrano, "A new transparent bridge protocol for lan internetworking using topologies with active loops," in ICPP '98: Proceedings of the 1998 International Conference on Parallel Processing. Washington, DC, USA: IEEE Computer Society, 1998, pp. 295–303.

[GHS83] R. Gallagher, P. Humblet, P. Spira, "A Distributed Algorithm for Minimum-Weight spanning trees", ACM Transactions Programming LAN and Systems, Jan 1983, pp. 66-77.

[GJK+09] Greenberg, A., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D., Patel, P., Sengupta, S., "VL2: A Scalable and Flexible Data Center Network", In Proceedings of ACM SIGCOMM, 2009.

[Had01] I. Hadzic. Hierarchical MAC address Space in Public Ethernet Networks, IEEE Globecom, 2001. p. 1563-1569.

[Har88] J. Hart, "Extending the IEEE 802.1 MAC bridge standard to remote Bridges," IEEE Netw. 2(1), 10–15 (1988).

[Har89] J. Hart, "Distributed Load Sharing", U.S. Patent 4.811.337, March 1989, http://www. freepatentsonline. com/4811337.html

[HMG06] M Huynh, P. Mohapatra, M. Goose, "Cross-Over Spanning Trees", Enhancing Metro Ethernet Resilience and Load Balancing, CSE 2006.

[IA04] G. Ibáñez, A. Azcorra, "Application of Rapid Spanning Tree Protocol for Automatic Hierarchical Address Assignment to Bridges", 11th International Telecommunication Networks Strategy and Planning Symposium, Networks 2004, Wien, June 2004.

[Iba+08a] G. Ibáñez et al., “ABridges: Scalable, self-configuring Ethernet campus networks”, Computer Networks Vol. 52, Issue 3, Feb. 2008, pp. 630-649.

[Iba+08b] G. Ibáñez et al., “Hierarchical Up/Down Routing Architecture for Ethernet backbones and campus networks”, HSN 2008, INFOCOM, April 2008.

[Iba09] G. Ibáñez et al., “Fast Path Bridges”, Disponible en [Internet]: http://ww.ieee802. org/1/ files/public/docs2009/fyi-ibanez-fast path-0909-02.pdf

[ICG+09] G. Ibáñez, J.A. Carral, A. García-Martínez y A. Azcorra, “Evolución conceptual de los protocolos de puentes tranparentes”, Novática, nº 198, marzo-abril 2009, pp. 55-62.

[ICG+10] Ibáñez, G.; Carral, J.A.; García-Martínez, A.; Arco, J.M.; Rivera, D.; Azcorra, A.; , "Fast Path Ethernet Switching: On-demand, efficient transparent bridges for data center and campus networks," Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on , vol., no., pp.1-7, 5-7 May 2010.

[ICR+11] Guillermo Ibáñez, Juan A. Carral, Diego Rivera, Aarón Montalvo, “ARP Path: ARP-based Shortest Path Bridges”, IEEE Communication Letters, July 2011.

[ICR+13] Guillermo Ibáñez, Juan A. Carral, Elisa Rojas, Jose Manuel Giménez-Guzmán, “Evaluating Native Load Distribution of ARP-Path Bridging Protocol in Mesh and Data Center”, aceptado para publicación en ICC’13, Budapest, junio 2013.

[IGA04] G. Ibáñez, A. García, A. Azcorra, “Alternative Multiple Spanning Tree Protocol (AMSTP) for Optical Ethernet Backbones”, En Proceedings of IEEE HSLN (LCN 2004) Tampa, Nov. 2004, Disponible en [Internet]: www.ieee.org/ieee.explore (Diciembre 2004).

[IGC+09] Ibanez, G., Garcia-Martinez, A., Carral, J.A., Arco, J., Azcorra, A., "Evaluation of tree-based routing ethernet", Communications Letters, IEEE, On page(s): 444 - 446 Volume: 13, Issue: 6, June 2009.

[IGC+10] Guillermo Ibanez, Alberto Garcia-Martinez, Juan A. Carral, Pedro A. Gonzalez, Arturo Azcorra, and Jose M. Arco. 2010. HURP/HURBA: Zero-configuration hierarchical Up/Down routing and bridging architecture for Ethernet backbones and campus networks. Comput. Netw. 54, 1 (January 2010), 41-56.

[Ish+04] K. Ishizu et al., “APG-Report: SSTP: An 802.1s Extension to Support Scalable Spanning Tree for Mobile Metropolitan Area Network”, In Proceedings of Globecom 2004, December 2004.

[ISN+11] Guillermo Ibáñez, Bart De Schuymer, Jad Naous, Diego Rivera, Elisa Rojas, Juan A. Carral, “Implementation of ARP-path low latency bridges in Linux and OpenFlow/NetFPGA”, High Performance Switching and Routing, Cartagena, July 2011.

[Iwa+04] A. Iwata et al. “Global Open Ethernet Architecture for a Cost-Effective Scalable VPN Solution,” IEICE Trans. On Communications, E87-B, 1, pp.142-151, Jan. 2004.

[Jai92] R. Jain, “A comparison of hashing schemes for address lookup in Computer Networks”, IEEE Transactions on Communications, vol. 40, no 3, pp. 1570-1573, Oct. 1992

[KCR08] C. Kim, M. Caesar, and J. Rexford, “Floodless in SEATTLE: A Scalable Ethernet Architecture for Large Enterprises”, In ACM SIGCOMM 2008, Aug. 2008.

[KDM09] Kvalbein, A., Dovrolis, C., Muthu, C., "Multipath load-adaptive routing: putting the emphasis on robustness and simplicity," Network Protocols, 2009, ICNP 2009, 17th IEEE International Conference on , vol., no., pp.203-212, 13-16 Oct. 2009.

[Kem86] M.F. Kempf. "Bridge Circuit for Interconnecting Networks". U.S. Patent 4.597.078, Junio 1986.

[Kle75] L. Kleinrock, "Queueing Systems. Volume 1: Theory", John Wiley & Sons, (Jan 2, 1975).

[KS01] G. Kuo and K.C. Shu. "Design of global hierarchical routing architecture on future IPv6 Internet". En Proceedings of Global Telecommunications Conference, 2001, GLOBECOM '01, vol. 1, pp. 121 – 125, Nov. 2001.

[KS06] P. Klein and N. Sprecher, "Provider Ethernet VLAN Cross Connect," Vol. January 2006. http://ww.ieee802.org/files/public/docs2006/new-sprecher-vlan-xc-ieee-0106.pdf

[KSG+09] Kandula, S., Sengupta, S., Greenberg, A., Patel, P., and Chaiken, R., "The Nature of Data Center Traffic: Measurements & Analysis", In Proceedings ACM IMC 2009.

[LG91] Y.-D. Lin and M. Gerla, "BRouter: The Transparent Bridge with Shortest Path in Interconnected LANs", En Proceedings of LCN, 1991, pp.175-183.

[LLN02] K.-S. Lui, W. C. Lee, and K. Nahrstedt, "STAR: A transparent spanning tree bridge protocol with alternate routing". En ACM SIGCOMM Computer Communications Review, vol. 32, Julio 2002.

[Lui02] K.S. Lui, "Alternate Routing Protocols for Bridged Networks". Ph.D. Thesis. University of Urbana-Campaign. 2002.

[Mej+06a] A. Mejía et al. "Incrementando las Prestaciones de Ethernet Usando Segment-Based Routing", Proceedings of XVII Jornadas de paralelismo. Albacete, September 2006.

[Mej+06b] A. Mejia, Flich, J., Duato, J., Reinemo, S.A., Skeie, T., "Segment-based routing: An efficient fault-tolerant routing algorithm for meshes and tori", In Proceedings of the 20th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006), IEEE (2006)

[MEZ04] A. Myers, T.S. Eugene, H. Zhang, "Rethinking the Service Model: Scaling Ethernet to a Million Nodes", En Proceedings of HOTNETS III. November 2004.

[MI06] J. Morales, G. Ibáñez, "Ethernet Fabric Routing (EFR): A scalable and secure ultrahigh speed switching architecture", High Speed Networking Workshop TCHSN INFOCOM 2006, Barcelone, April 2006.

[Moo65] Moore, Gordon E., "Cramming more components onto integrated circuits", Electronics, Volume 38, Number 8, April 19, 1965.

[MR09] "MapReduce. The Programming Model and Practice", Sigmetrics 2009, Tutorials.

[MRP] "IEEE Standard for Local and Metropolitan Area Networks Virtual Bridged Local Area Networks-Amendment 07: Multiple Registration Protocol", Disponible en [Internet]: < http://www.ieee802.org/1/pages/ 802.1ak.html>, Julio 2005.

[Mys+09] Radhika Niranjan Mysore, Andreas Pamboris, Nathan Farrington, Nelson Huang, Pardis Miri, Sivasankar Radhakrishnan, Vikram Subramanya, and Amin Vahdat. "PortLand: a scalable fault-tolerant layer 2 data center network fabric", SIGCOMM Computer Communications Review 39, 4 (August 2009), 39-50.

[MZ96] B. McDonald, T. Znati, “Comparative analysis of neighbor greeting protocols. ARP versus ES-IS”, en Proceedings of IEEE 29th Annual Simulation Symposium, April 1996.

[NetFPGA] NetFPGA http://www.netfpga.org/

[NRS] NRS Reference Networks, http://www.ibcn.intec.ugent.be/INTERNAL/ NRS/index.html

[OFS] “OpenFlow Switches”, Disponible en [internet]: http://www.openflowswitch.org/

[OMNET] “Omnet++ Simulator”, Available on line: http://www.omnetpp.org

[PBBT] IEEE 802.1Qay Provider Backbone Bridge Traffic Engineering. http://www.ieee802.org/1/pages/802.1ay.html

[PD08] Prasad, R.S., Dovrolis, C. , "Beyond the Model of Persistent TCP Flows: Open-Loop vs Closed-Loop Arrivals of Non-persistent Flows," Simulation Symposium, 2008. ANSS 2008. 41st Annual , vol., no., pp.121-130, 13-16 April 2008.

[Pel+04] F. D. Pellegrini, D. Starobinski, M. G. Karpovsky, and L. B. Levitin, “Scalable cycle-breaking algorithms for gigabit Ethernet backbones”, En Proceedings of IEEE Infocom 2004, Marzo 2004.

[Pel+06] V. Pellegrini et al., “Scalable, “Distributed cycle-breaking algorithms for gigabit Ethernet backbones”, Journal of Optical Networking, Vol. 5, No. 2, Feb. 2006, pp. 122-144.

[Per00] R. Perlman, “Interconnections. Bridges, Routers, Switches, and Internetworking Protocols. Second Edition”, Addison-Wesley, 2000.

[Per04] R. Perlman, “RBridges: Transparent routing,” in Proceedings of IEEE Infocom 2004, March 2004.

[Per85] R. Perlman, “An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN”, En Proceedings of Ninth ACM Data Communications Symposium, Vol. 20, No. 7, p. 44–52, Septiembre 1985, New York, USA.

[PTY04] R. Perlman, J. Touch, A. Yegin, “RBridges: Transparent Routing”, Disponible en [Internet]: http://www.ietf.org/internet-drafts/draft-perlman-RBridge-00.txt. April 2004.

[PYTHON] Python Programming Language – Official Website: http://www.python.org/

[RBridge] “The RBridge archives”, Disponible en [Internet]: http://www.postel.org/pipermail/RBridge/

[RF91] B. Rajagopalan, M. Faiman, “Load sharing and shortest-path routing in transparently interconnected local area networks”, Proceedings Tenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1991, IEEE 7-11 April 1991. Vol. 3, pp. 1135 – 1144.

[RFC2992] C. Hopps, “Analysis of an Equal-Cost Multi-Path Algorithm”, November 2000.

[RFC3561] C. Perkins, E. Belding-Royer, “Ad hoc On-Demand Distance Vector (AODV)”, July 2003.

[RFC826] D. C. Plummer, “An Ethernet address resolution protocol”, Internet Engineering Task Force. Request for Comments: 826. Noviembre 1982.

[RS91] Thomas L. Rodeheffer and Michael D. Schroeder, “Automatic reconfiguration in Autonet”, SIGOPS Oper. Syst. Rev. 25, 5 (September 1991), pp. 183-197.

[RTA00] T. L. Rodeheffer, C. A. Thekkath, and D. C. Anderson, “Smartbridge: a scalable bridge architecture,” SIGCOMM Computer Communication Review, vol. 30, no. 4, pp. 205–216, 2000.

[Rut09] R. Sofia, “A survey of advanced Ethernet forwarding approaches”, Communications Surveys & Tutorials, IEEE. Vol.11, no.1. pp. 92-115, 2009. DOI:102209/SURV.2009.090108

[SC88] W. D. Sincoskie, C.J. Cotton, “Extended bridge algorithms for large networks”, IEEE Network, Volume 2, Issue 1, Jan. 1988, pp. 16 – 24

[Sch+91] M. Schroeder et al., “Autonet: A High-Speed, Self–Configuring Local Area Network Using Point –to–Point Links”, IEEE Journal on Selected Areas in Communications, Vol. 9, No. 8, pp. 1318–1335, October 1991.

[SD97] F. Silla and J. Duato, “On the Use of Virtual Channels in Networks of Workstations with Irregular Topology”, En Proceedings of the 1997 Parallel Computing, Routing, and Communication Workshop, Junio 1997, Lecture Notes In Computer Science Vol. 1417.

[Sea99] M. Seaman, “Loop cutting in the original and rapid spanning tree algorithms”, IEEE, Noviembre 1999. Disponible en [Internet]: <http://www.ieee802.org/1/files/public/ docs999/ loop_cutting08.pdf/ >

[Sei00] R. Seifert, “The All-New Switch Book: The complete guide to LAN switching technology”, John Wiley & Sons, 2008.

[Sei88] William M. Seifert, “Bridges and Routers”, IEEE Network, January 1988.

[Sha+04] S. Sharma, K. Gopalan, S. Nanda, and T. C. Chiueh, “Viking: a multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks,” in INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Society (IEEE, 2004), pp. 2283–2294, Mars 2004.

[SIMPY] SimPy Simulation Package (Python) – Official website: http://simpy.sourcefprge.net

[SKZ02] L. Starobinski, M.G. Karpovsky, L. Zakrevski, “Application of Network Calculus to General Topologies using Turn-Prohibition”, IEEE INFOCOM 2002 p. 1151-1159. 0-7803-7476-2/02.

[SL02] T. Skeie, O. Lysne, “Layered shortest path (LASH) routing in irregular system area networks”, Proceedings of Parallel and Distributed Processing Symposium, IPDPS April 2002, pp 162-169.

[SP88] M. Soha, R. Perlman, “Comparison of two LAN bridge approaches”, Network, IEEE Volume 2, Issue 1, pp. 37–43, Jan. 1988

[SRF+02] J. C. Sancho, A. Robles, J. Flich, P. Lopez and J. Duato, “Effective Methodology for Deadlock-Free Minimal Routing Networks”, ICPP '02 Proceedings of the 2002 International Conference on Parallel Processing.

[TG91] T.-Y. Tai, M. Gerla, “LAN Interconnection: A Transparent, Shortest-Path Approach”, En Proceedings ICC '91, 1991. pp. 1666-1670.

[Y1731] Recommendation ITU-T Y.1731 - OAM functions and mechanisms for Ethernet based networks, 07/11.