Impacto de los túneles MPLS en la topología de Internet

Tesis de Maestría

Autor:
Ing. Fernando Gabriel Dávila Revelo

Director:
Dr. Ing. José Ignacio Alvarez-Hamelin

Tesis realizada como parte de los requisitos
para la obtención del título de
Master en Ingeniería en Telecomunicaciones

Grupo de redes complejas y comunicación de datos
Facultad de Ingeniería

Diciembre de 2014

UNIVERSIDAD DE BUENOS AIRES

Resumen

Grupo de redes complejas y comunicación de datos
Facultad de Ingeniería

Master en Ingeniería en Telecomunicaciones

Impacto de los túneles MPLS en la topología de Internet

por Ing. Fernando Gabriel Dávila Revelo

El presente trabajo estudia el comportamiento de los enlaces *MPLS* en la topología de Internet. Por un lado evalua los métodos para identificar los *LSRs* dentro de un *LSP* y por otro presenta un análisis basado en la teoría de grafos para comprender mejor las características de las redes *MPLS* y su interacción con las redes *IP* en Internet.

Índice general

Índice de figuras

Índice de tablas

Abreviaturas

APAR	**A**nalitical and **P**robe-based **A**lias **R**esolver
AS	**A**utonomous **S**ystem
BGP	**B**order **G**ateway **P**rotocol
IANA	**I**nternet **A**ssigned **N**umber **A**uthority
ICMP	**I**nternet **C**ontrol **M**essage **P**rotocol
IGMP	**I**nternet **G**roup **M**anagement **P**rotocol
IP	**I**nternet **P**rotocol (address)
LER	**L**abel **E**dge **R**outer
LSE	**L**abel **S**tack **E**ntries
LSP	**L**abel **S**witching **P**ath
LSR	**L**abel **S**witching **R**outer
MPLS	**M**ulti-**P**rotocol **L**abel **S**witching
TCP	**T**ransmission **C**ontrol **P**rotocol
TOS	**T**ype **o**f **S**ervice
TTL	**T**ime **t**o **L**ive
UDP	**U**ser **D**atagram **P**rotocol
VPN	**V**irtual **P**rivate **N**etwork

Capítulo 1

Introducción

El estudio de la topología de Internet surge de la necesidad de obtener redes artificiales que permitan simular el comportamiento de Internet. Es decir, obtener modelos que representen con la mayor precisión posible a la red real, a fin de poner a prueba nuevos protocolos y algoritmos de comunicaciones; por ejemplo, protocolos de enrutamiento.

Sin embargo las motivaciones no se quedan en justificaciones tecnológicas solamente, posteriormente el estudio de la topología de Internet ha permitido revelar cómo las decisiones tecnológicas, políticas o económicas impactan en las diferentes jerarquías de la conectividad de Internet. Por ejemplo, permite relevar las interconexiones existentes entre proveedores, información que es usualmente confidencial. Asimismo, se puede determinar el posicionamiento y relevancia de un *carrier* en base a su ubicación dentro del grafo de Internet. Por otro lado, en el ámbito de ingeniería, conocer con precisión la topología de Internet permite determinar la ubicación más eficiente para desplegar nuevos servicios, o las acciones estratégicas necesarias para cumplir con parámetros de Calidad de Servicio.

La necesidad de conocer la topología de Internet trae implícito consigo otras exigencias propias del desarrollo tecnológico, por ejemplo identificar enlaces *IPv4*, enlaces *IPv6* y recientemente (2011) han comenzado los estudios para cuantificar los enlaces *MPLS* [SBE11] [Don13]. Sin embargo y a pesar de que el despliegue de *MPLS* no es reciente y su proceso de estandarización comienza en 1997 [Cal+97], se conoce poco sobre el comportamiento y el impacto de *MPLS* en Internet. Actualmente no solamente hace falta una cuantificación de la presencia de enlaces *MPLS*, sino que además es necesario

estudiar por ejemplo, las características y la ubicación de estos enlaces, la frecuencia con la cual un nodo *IP* se conecta a un nodo *MPLS*, las propiedas de las redes *MPLS* y su interacción con redes *IPv4*, etc. Estas características entre otras, son necesarias tanto para evaluar los efectos de las redes *MPLS* como para entender mejor el despliegue de esta tecnología.

En el presente trabajo se utiliza un software de exploración de redes basado en *Python* y un mecanismo usando la bibliografía existente para inferir los túneles *MPLS* a partir de los datos recolectados. A partir de esta información se propone además un método para construir diversos grafos donde se identifiquen los nodos y las redes *MPLS* con el objeto de conocer con mayor detalle sus características y su impacto en la topología típica de *Internet*.

1.1. Organización de la Tesis

El presente trabajo se desarrolla en tres capítulos: Estado del Arte (Capítulo 2), Experiencias y Análisis de las exploraciones (Capítulo 3) y Conclusiones (Capítulo 4). Los temas de cada capítulo se presentan en el mismo orden que surgieron cuando se desarrolló la Tesis.

En el Capítulo 2 se describe la situación actual del tema de investigación y los fundamentos teóricos sobre los cuales se desarrolló el trabajo, se presenta una introducción a los distintos modelos de topología y se hace hincapié en la topología de *routers* discutiendo sus limitaciones. Entre estas limitaciones se explican minuciosamente los problemas alrededor de la Resolución de Alias y de la identificación de túneles *MPLS*, pues son la principal base teórica para el desarrollo posterior.

En el Capítulo 3 se expone el trabajo realizado. Aquí se describe de forma paralela tanto el desarrollo del trabajo como el análisis alrededor de los resultados obtenidos

Finalmente el Capítulo 4 concluye la presentación sintetizando el trabajo realizado.

Capítulo 2

Estado del Arte

El presente capítulo presenta una descripción de la situación actual del tema basada en la literatura existente hasta la fecha. De este modo inicialmente se describen los distintos niveles de la Topología de Internet y las herramientas principales para la recopilación de datos. Posteriormente, de entre todas las topologías descritas, al ser la Topología a nivel de *routers* la que motiva este estudio, se profundiza en ella. Se introduce también las limitaciones de la Topología a nivel de *routers* haciendo hincapié en la Resolución de Alias y los túneles *MPLS*. Finalmente, se describen las métricas empleadas para el análisis de grafos y las herramientas disponibles para ello.

2.1. Topología del Internet

La presente sección presenta una breve descripción de los diferentes niveles en los cuales se estudia la Topología de Internet: nivel *IP*, nivel de *routers*, y nivel de Sistemas Autónomos (*AS*). Se describen también las herramientas más usadas para la recopilación de datos de cada una de las topologías y los proyectos más relevantes que en la actualidad se llevan a cabo.

2.1.1. Topología a nivel *IP*

La Topología de Internet a nivel *IP* corresponde a un grafo construido usando típicamente las salidas de la herramienta *traceroute* en donde las interfaces *IP* se consideran

como nodos y las aristas se forman entre aquellas direcciones *IP* que presenten adyacencias a la salida del *traceroute*. En esta topología cada *router* aparece separado en tantos nodos como interfaces *IP* se descubran, si los caminos son disjuntos.

Actualmente existen varios proyectos dedicados a realizar exploraciones de Internet mediante *traceroutes* que permiten obtener topologías a nivel *IP*. Algunos de los principales proyectos ([HFc12]) son: *DIMES*, *iPlane* y *Ark IPv4 All Prefix /24*. Más información sobre otros proyectos puede encontrarse en [Don13].

DIMES[1] es un proyecto llevado a cabo por la Universidad de Tel Aviv. Se basa en lanzar varios *traceroutes* en paralelo. Estas sondas son lanzadas por usuarios que han instalado el *software netDIMES* de forma voluntaria. Al día de hoy tiene más de 9600 usuarios distribuidos. *DIMES* no publica toda la información recopilada a través los *traceroutes*, en lugar de ello publica directamente las adyacencias *IP* descubiertas.

iPlane[2] por otro lado, es un proyecto desarrollado por la Universidad de Washington con alrededor de 250 puntos de medición utilizando la infraestructura de *PlanetLab* (infraestructura que también se usó para el desarrollo de este trabajo), y permite construir un mapa de la Topología *IP* de Internet con el objetivo de predecir el rendimiento de *punta a punta* entre dos nodos cualesquiera, partiendo de ciertos enlaces cuyo rendimiento es previamente conocido. Típicamente los enlaces conocidos se tratan de enlaces altamente transitados ubicados en la troncal de Internet.

Finalmente *Ark IPv4 All Prefix /24*[3] es un proyecto llevado a cabo por *CAIDA* que consiste en la recolección de datos desde 54 servidores dedicados que envían sondas de prueba a una *IP* seleccionada de forma aleatoria de entre cada red IPv4/24 activa en Internet. La sonda de prueba de *Ark IPv4 All Prefix /24* se basa en la herramienta *paris-traceroute*[4] en lugar del *traceroute* tradicional.

2.1.2. Topología a nivel *router*

El segundo nivel de la Topología de Internet es a nivel de *routers*. En esta topología cada *router* corresponde a un nodo del grafo y las aristas se forman entre los *routers* que

[1] http://www.netdimes.org/about.html
[2] http://iplane.cs.washington.edu/
[3] http://www.caida.org/projects/ark/
[4] http://www.paris-traceroute.net/

se encuentren conectados entre sí. Los datos para formar este tipo de grafos se extraen típicamente mediante *traceroutes* y más recientemente mediante sondas *IGMP* [Mé+09].

Al emplear los datos obtenidos mediante la herramienta *traceroute*, es necesario identificar las interfaces *IP* descubiertas que forman parte de un mismo *router*, es decir: inicialmente se obtiene un grafo equivalente a uno de nivel *IP* y posteriormente las interfaces *IPs* correspondientes a un mismo *router* se contraen en un único nodo, este proceso es conocido como Resolución de Alias.

Por otro lado, la exploración empleando sondas *IGMP* consiste en recolectar todas las interfaces *multicast* de un *router* mediante una única exploración, con lo cual el proceso de Resolución de Alias ya no es necesario. Básicamente el proceso consiste en enviar desde un monitor mensajes *IGMP* `ask-neighbor`. Posteriormente este mensaje es contestado por cada uno de los *routers* consultados con un mensaje *IGMP* `neighbor-reply` en donde se incluyen todas las interfaces multicast locales habilitadas e información de los vecinos conectados a cada interfaz. Entre los proyectos basados en descubrir la Topología de Internet a nivel de *routers* empleando sondas *IGMP* están: *mrinfo*[5] y *mrinfo-rec* [MDBP10]. Sin embargo se ha encontrado que los grafos resultantes de este tipo de exploraciones son incompletos e incluso presentan componentes desconexas debido al progresivo crecimiento de filtros *IGMP* en Internet [Don13] [Mar+12]. Para contrarrestar estas deficiencias *Marchetta et al.* han propuesto *MERLIN*, una herramienta híbrida que además de sondas *IGMP* emplea sondas basadas en *traceroutes* que posteriormente son procesadas con técnicas de Resolución de Alias [Mar+11].

2.1.3. Topología nivel Sistema Autónomo

Un *AS* es una red o conjunto de redes que funcionan bajo una misma administración, usualmente: un ISP, una universidad, o una empresa. Cada *AS* se identifica mediante un único número asignado por la *Internet assigned number authority (IANA)*. Un *AS* también puede definirse de forma más práctica como un conjunto de *routers* bajo el mismo dominio.

La Topología nivel Sistema Autónomo de Internet no es más que un grafo en donde un nodo está formado por un *AS* o en otras palabras por un conjunto de *routers* que forman

[5] `http://cvsweb.netbsd.org/bsdweb.cgi/src/usr.sbin/mrinfo/`

parte de un misma entidad administrativa y las aristas se forman por las conexiones existentes entre los distintos *ASs*.

Este nivel de topología permite conocer no solamente el comportamiento de las conexiones existentes entre distintas redes, sino además permite conocer a través de las relaciones entre *ASs*, los distintos tipos de acuerdos comerciales existentes entre ISPs, por ejemplo, quién es cliente y quién es proveedor. Por tanto, el grafo resultante de esta topología es típicamente un grafo dirigido que brinda información importante sobre el ruteo en Internet, así como en políticas de ruteo y acuerdos comerciales [TGSE01].

Las relaciones entre *ASs* se pueden agrupar en tres categorías generales: *cliente a proveedor* (c2p del inglés *customer-to-provider*), *par a par* (p2p del inglés *peer-to-peer*) y *socio a socio* (s2s del inglés *sibling-to-sibling*). En la relación c2p un *AS* cliente paga a un *AS* proveedor por todo tráfico existente entre los dos. En la relación p2p, dos *ASs* intercambian sin remuneración económica el tráfico generado por ellos y por sus *ASs* clientes. Finalmente en la relación s2s, dos ASs pertenecientes a una misma organización intercambian sin remuneración económica cualquier tipo de tráfico existente entre ellos, ya sea tráfico de sus proveedores, clientes u otros *socios* [Dim+07a].

Es importante notar que en la Topología a nivel *AS* el grafo es dirigido, a diferencia de las topologías previamente descritas. Esta problemática de crear grafos dirigidos surge de la necesidad de inferir el tipo de relación entre dos *ASs* y es ampliamente estudiada. Se puede encontrar más información al respecto en [Gao01], [DB+07], [XG04], [Dim+07b]

Con respecto a las fuentes de datos, para la construcción de este tipo de grafos se tiene las tablas BGP y bases de datos públicas.

Los *ASs* intercambian rutas mediante *Border Gateway Protocol (BGP)* [RL95]. Cada *router* que habla BGP elabora una tabla que contiene un *path* de *ASs* hacia cada red destino, con esa información se construye la aristas del grafo. Algunos proyectos que recopilan tablas BGP de Internet son *Route-Views*[6] o *RIPE NCC*[7]. Estos proyectos se basan en capturar los estados de las tablas BGP de varios *routers* con los cuales se conectan periódicamente.

Otra opción de obtener información para crear esta topología es usar información disponible en bases de datos públicas de organizaciones responsables de registrar las *IPs* y

[6] http://www.routeviews.org/
[7] http://www.ripe.net/data-tools/stats/ris/routing-information-service/

ASs. Estas organizaciones tienen por objeto facilitar la administración de Internet y información se puede obtener de *Internet Regional Registry*[8] o *Internet Routing Registry*[9].

2.2. Limitaciones en la Topología a nivel de *routers*

En esta sección se describen los problemas y las soluciones existentes para la obtención de la Topología a nivel de *routers*. Inicialmente se describe la problemática de detectar falsos enlaces debido a los balanceadores de carga cuando se usa el *traceroute* tradicional para la exploración de datos. Posteriormente se exponen las recomendaciones para procesar la información, considerando que al recopilar los datos existen interfaces que no responden y los problemas para Resolución de Alias. Finalmente se detalla una de las problemáticas recientemente estudiadas que podría afectar a la Topología a nivel de *routers* y que es objeto del presente trabajo: los túneles *MPLS*.

2.2.1. Falsos Enlaces y Balanceadores de Carga

La herramienta más común para estudiar la Topología de Internet es la herramienta *traceroute*. Esta herramienta consiste básicamente en enviar una serie de mensajes *ICMP* `echo-request` hacia el destino y mediante el campo *TTL* (*Time To Live*) descubrir los *routers* intermedios existentes en la ruta.

Partiendo de que el *TTL* del mensaje *ICMP* `echo-request` se decrementa en 1 una vez que pasa por un *router*, el *traceroute* se explica del siguiente modo: Se envían mensajes al destino aumentando el valor del *TTL* de forma gradual y empezando por el valor de 1. El primer *router* recibe el mensaje, decrementa el *TTL* y enseguida lo descarta por haber alcanzado el valor 0. Al momento de descartarlo envía un mensaje *ICMP* `time-exceeded` al origen. El siguiente mensaje enviado tiene un *TTL* con valor 2, el primer *router* decrementa el *TTL* como lo hizo previamente pero esta vez deja pasar el mensaje por que el *TTL* aún no expira, siendo el segundo *router* quien descartará el mensaje y enviará un *ICMP* `time-exceeded` al origen. Este proceso se repite atravesando todos los *routers* intermedios hasta que el destino es alcanzado y responde con un *ICMP* `echo-reply`.

[8] http://www.isoc.org/briefings/021/
[9] ftp://ftp.radb.net/routing.arbiter/radb/dbase/

Además de mensajes *ICMP*, el *traceroute* puede funcionar usando Datagramas *UDP*, o paquetes *TCP* (*tcptraceroute*). Típicamente dependiendo del sistema operativo la herramienta permite al usuario elegir entre estos 3 protocolos.

El *traceroute* es una herramienta útil para diagnosticar en qué segmento de la red se encuentra un determinado problema, sin embargo no es una herramienta que permite conocer el camino real que toman los paquetes hasta llegar a su destino.

En Internet los *routers* pueden enviar el tráfico a través de diferentes caminos, a este proceso se le llama balanceo de carga y puede ser: por flujo, por paquetes o por destino [Tha00].

El balanceo de carga por flujo se basa en asignar a un mismo flujo todos los paquetes que contengan el mismo *identificador de flujo*, siendo cada flujo enviado siempre por una misma interfaz del *router*. El *identificador de flujo* que permite esta clasificación está compuesto por los siguientes campos del encabezado *UDP* y *TCP*: Dirección *IP* origen, Dirección *IP* destino, Protocolo, Puerto origen y Puerto destino; y otros tres campos propios del encabezado *IP* e *ICMP*: Tipo de Servicio *IP* (TOS), código *ICMP* y *Checksum* [CIS] [JUN].

Por otro lado, el balanceo por paquetes tiene como objetivo solamente la distribución de carga del *router* sin importar ningún otro aspecto, con lo cual dos paquetes aun perteneciendo al mismo flujo podrían ser enviados por distintas interfaces del *router*.

Finalmente la política de balanceo de carga por destino envía el paquete basado directamente en la *IP* destino, es decir no hay un balanceo de carga *per se* y se trata al paquete de acuerdo al enrutamiento tradicional.

Las políticas de balanceo de carga expuestas previamente hacen que todos los mensajes enviados por un mismo *traceroute* puedan seguir caminos distintos, lo cual produce que se descubran falsos enlaces y otros nunca sean descubiertos. Los falsos enlaces podrían descubrirse de manera aleatoria dependiendo del tipo de balanceador de carga que atraviesa cada paquete. Este problema se explica en la figura 2.1. En el ejemplo se tiene un balanceador L a 1 salto de distancia del origen del *traceroute* y 5 *routers* representados por un círculo. A la izquierda se observa la topología real y las flechas amarillas indican el *router* en donde expiró el *TTL* de cada mensaje *ICMP* del *traceroute*. Siguiendo la ruta de los mensajes *ICMP* se inferiría la topología de la derecha, en donde debido al

balanceador de carga algunos enlaces no se descubrieron (*L-B*, *B-D*, *A-C*, *C-E*) y otros se descubrieron erróneamente (*A-D*). Aumentando el número de paquetes enviados por *traceroute* es posible disminuir el número de enlaces que no se descubren pero los falsos enlaces siempre estarán presentes.

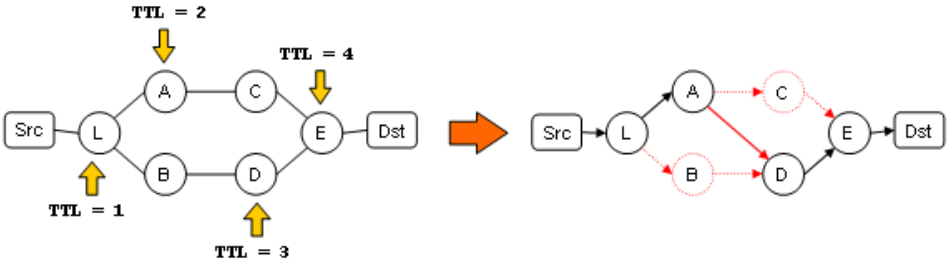


FIGURA 2.1: Efecto de los balanceadores de carga en la detección de falsos enlaces[10].

Los falsos enlaces son producto de la existencia de políticas de balanceo de carga por flujo y por paquetes.

Esta inferencia de falsos enlaces debido al balanceo de carga de los *routers* es una importante limitación del *traceroute* tradicional. Como alternativa se ha desarrollado una herramienta que soluciona estos inconvenientes denominada *paris-traceroute*.

Paris-traceroute permite que todos los paquetes enviados a un mismo destino pertenezcan a un mismo flujo, con lo cual los balanceadores de carga por flujo no producen falsos enlaces y todos los paquetes siguen un mismo camino. Para esto *paris-traceroute* mantiene constante todos los campos involucrados en asociar el paquete a un flujo determinado.

En la versión de *traceroute* tradicional y *tcptraceroute* es necesario que cada paquete de prueba pueda ser identificado dentro de todos los paquetes que un mismo *traceroute* puede generar, para ello se usa un identificador en cada paquete que varía de acuerdo al tipo de protocolo que se emplee: En los mensajes *ICMP* se usa *Sequence Number* como identificador, en los mensajes *UDP* el identificador es el *Destination Port* y en los mensajes *TCP* el identificador es el campo *Identification* del encabezado *IP*.

Paris-traceroute tiene como reto mantener constantes todos los campos involucrados en el balanceo de carga por flujo para que todos los paquetes de prueba pertenezcan

[10]Imagen extraída de http://www.paris-traceroute.net/

siempre al mismo flujo y por ende sigan siempre el mismo camino, pero al mismo tiempo debe permitir que cada paquete sea identificado de forma única dentro de todos los paquetes enviados en un mismo *traceroute* [Aug+06]. Para lograr esto, en el caso de los mensajes *ICMP* se usa el *Sequence Number* para identificar cada paquete al igual que en el *traceroute* tradicional, pero además se asigna al campo *Identifier* un valor que permita mantener constante el *checksum* en todos los paquetes, dando como resultado un mismo *identificador de flujo*. En los mensajes *UDP* se usa el *checksum* para identificar cada paquete en lugar del *Destination Port* como se hacía en el *traceroute* tradicional, pues el campo *Destination Port* forma parte del *identificador de flujo* y es necesario mantenerlo constante. Para variar el *checksum* se modifica el *payload* del paquete y se mantiene constante el resto del encabezado *UDP*. Finalmente en los mensajes *TCP*, *paris-traceroute* no hace modificaciones considerables al funcionamiento del *tcptraceroute* que ya mantenía constantes todos los campos que definen el *identificador de flujo*, pero usa el campo *Sequence Number* como identificador del paquete en lugar del campo *Identification*.

La Figura 2.2 muestra los campos de los encabezados *IP*, *UDP*, *ICMP* y *TCP* usados para el balanceo de carga por flujo. Se observa que en resumen, los campos que definen el *identificador de flujo* son los primeros 32 bits del *payload IP* y son precisamente estos campos los que *paris-traceroute* mantiene constantes.

Paris-traceroute permite evitar que los paquetes de prueba enviados sigan distintos caminos cuando atraviesan balanceadores de carga por flujo, evitando así que se descubran falsos enlaces. Los balanceadores de carga por destino no son problema puesto que todos los paquetes de prueba siempre seguirán el mismo camino al tener el mismo destino. Finalmente para los balanceadores de carga por paquetes, *paris-traceroute* no presenta ninguna solución, sin embargo se estima que solamente el 2 % de las rutas atraviesan este tipo de balanceadores y se cree que enviando 6 paquetes de prueba hacia cada *router* descubierto *hop-by-hop*, se puede asumir con un 95 % de confiabilidad que si las respuestas a cada uno de estas 6 pruebas llegan con el mismo *TTL*, entonces no se atravesó ningún balanceador de carga por paquetes[AFT07].

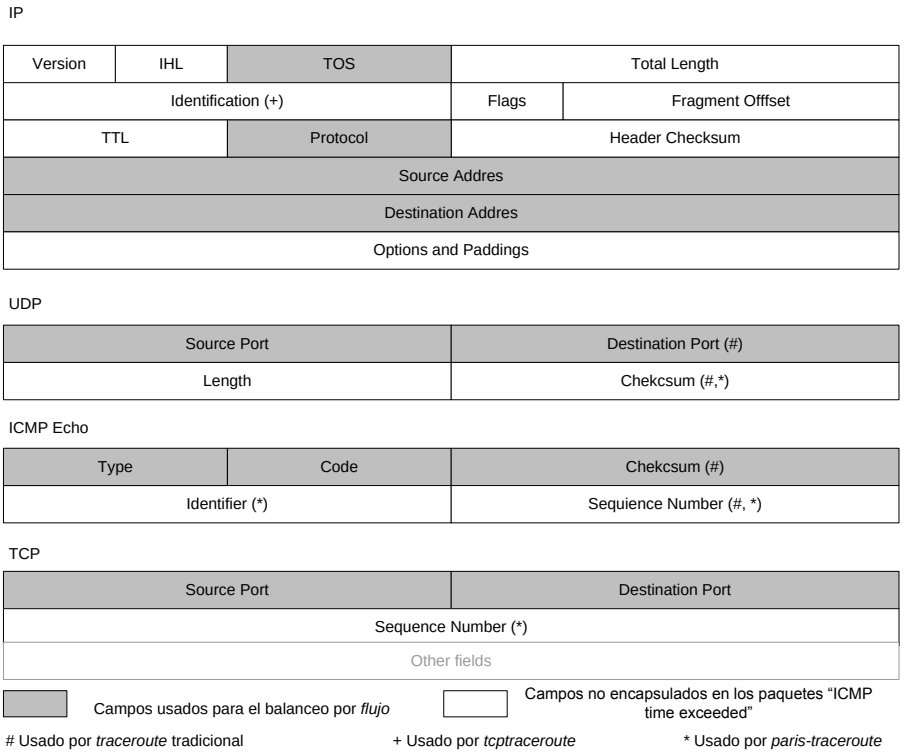


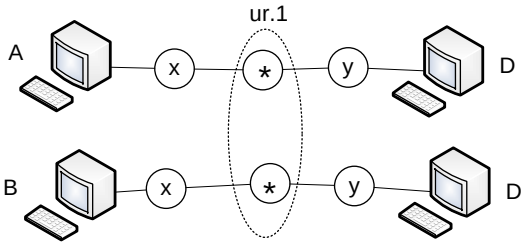
FIGURA 2.2: Campos involucrados en el balanceo de carga por flujo.

2.2.2. Interfaces que no responden

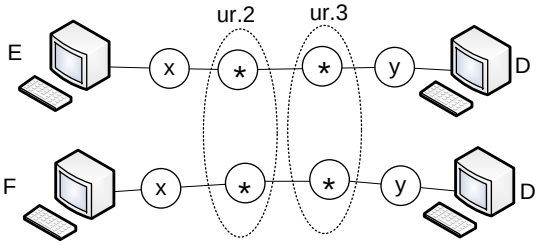
En Internet existen varios *routers* cuyas interfaces se han configurado para no responder ningún mensaje de control ante la expiración del TTL, mostrando un "$*$" como salida del *traceroute*. En [BSK06] se propone que para disminuir el número de interfaces que no responden, se pueden comparar los diversos caminos seguidos por los *traceroutes* con el objetivo de identificar aquellos "$*$" que podrían corresponder a una misma interfaz, de la siguiente forma:

i Sean dos *traceroutes* $t_{A->D}$ y $t_{B->D}$ que contienen un "$*$" entre dos *IPs* conocidas. Si la *IP* predecesora y sucesora al "$*$" son iguales en $t_{A->D}$ y $t_{B->D}$, y además tienen el mismo destino D, entonces se asigna un mismo nombre a la interfaz sin respuesta, ejemplo: *ur.1*. Este caso se ilustra en la Figura 2.3(a).

ii Sean dos *traceroutes* $t_{A->D}$ y $t_{B->D}$ que contienen dos "$*$" consecutivos entre dos *IPs* conocidas. Si la *IP* predecesora y sucesora al par de "$*$" son iguales en $t_{A->D}$ y $t_{B->D}$, y además tienen el mismo destino D, entonces se asigna un mismo nombre



(a) Una interfaz no responde



(b) Dos interfaces no responden

FIGURA 2.3: Interfaces que no responden.

a la primera interfaz sin respuesta y otro nombre a la segunda, ejemplo: *ur.1, ur.2.* Este caso se ilustra en la Figura 2.3(b).

iii Si un *traceroute* contiene más de dos interfaces sin respuesta consecutivas, este debe ser descartado.

La solución descrita y propuesta en [BSK06] considera que para resolver los *routers* que no responden se deben comparar los *traceroute* con el mismo destino, a esta condición se le llamará en el presente trabajo condición "Destino Común".

2.2.3. Resolución de Alias

Como se mencionó previamente en la sección 2.1.2, el proceso de Resolución de Alias consiste en identificar cuales de las interfaces obtenidas mediante una exploración tipo *traceroute* pertenecen a un mismo *router*. Este proceso puede realizarse mediante métodos activos o pasivos. El método activo consiste en enviar sondas basadas típicamente en paquetes *ICMP* y *UDP* que permitan obtener información extra para identificar las interfaces de un mismo *router*. El método pasivo que es el usado en el presente trabajo, consiste en realizar un procesamiento de la información ya obtenida a través de los *traceroutes* para descubrir los alias de una interfaz. Este método es denominado por sus autores como *APAR (Analytical and Probe-based Alias Resolver)* [GS09]. *APAR* se basa

en identificar los alias a partir de la recolección de datos obtenida mediante herramientas tipo *traceroute* sin usar sondas adicionales.

APAR básicamente se basa en identificar los enlaces punto a punto y los enlaces multi-acceso para inferir los alias *IP*.

2.2.3.1. Enlaces punto a punto

Partiendo de que en una subred de máscara /30 o /31, únicamente se permiten enlaces punto a punto, es decir que tienen un máximo de 2 interfaces *IP* conectadas, se pueden alinear los *traceroutes* obtenidos en la recolección de datos de forma que se puedan inferir adecuadamente los alias.

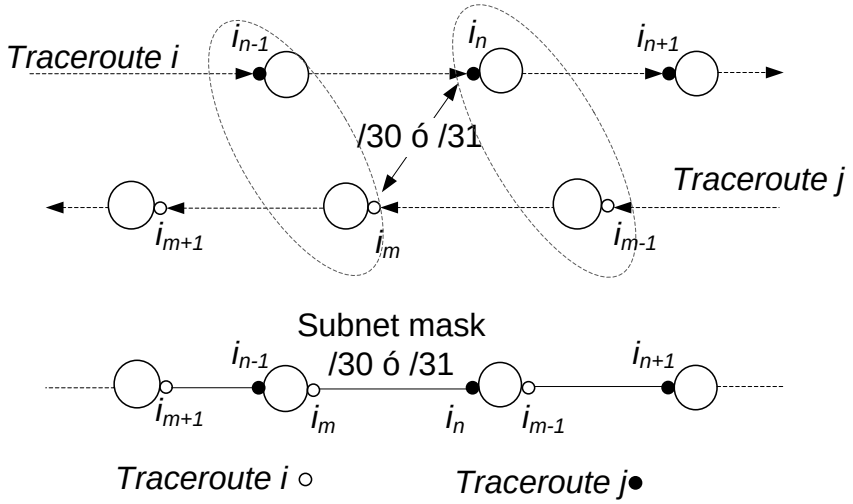


FIGURA 2.4: Resolución de Alias enlaces punto a punto. En la parte superior se muestra las interfaces descubiertas en los recorridos realizados por los *traceroutes* i y j. En la parte inferior se muestra el resultado inferido una vez resueltos los Alias.

Por ejemplo, sean dos interfaces i_n e i_m que pertenecen a una misma subred s de máscara /30 o máscara /31 y que aparecen en los *traceroutes* t_i y t_j respectivamente como se muestra en la Figura 2.4. Se puede afirmar por el valor de las máscaras de la subred que las interfaces están unidas por un enlace point-to-point, lo cual implica que la interfaz i_{n-1} precedente a i_n debe formar parte del mismo *router* que la interfaz i_m; y que la interfaz i_{m-1} debe formar parte del mismo *router* que la interfaz i_n. Se dice entonces que cada par de interfaces i_{n-1}, i_m e i_{m-1}, i_n forman Alias *IP*. Se llega a esta conclusión debido a que las interfaces i_n, i_m solamente pueden tener un nodo vecino en el sentido

contrario al del *traceroute*, ya que se partió de que estas interfaces forman un enlace punto a punto.

2.2.3.2. Enlaces multi-acceso

Los enlaces multi-acceso son usados para conectar diversas interfaces a una misma subred. Estas subredes incluyen más de 2 nodos conectados a ellas. Al igual que en los enlaces punto a punto, en este caso se pretende también encontrar las interfaces que pertenezcan a una misma subred y en base a esta información ordenar los *traceroutes* para inferir los alias *IP*. Sin embargo cuando el enlace es multi-acceso existen más de 2 nodos perteneciendo a la misma subred y esto ocasiona que no se puedan inferir los alias con la misma facilidad que en el caso de los enlaces punto a punto. La figura 2.5 muestra un esquema de red que ejemplifica la problemática de los enlaces multi-acceso, en la figura el *traceroute* t_i descubre las interfaces i_{m-1}, i_m, i_{m+1} y el *traceroute* t_j descubre las interfaces j_{n-1}, j_n, j_{n+1}, la alineación de los *traceroute* comienza identificando las interfaces que pertenezcan a la misma subred, en la figura las interfaces i_m, j_n pertenecen a la misma subred del enlace multi-acceso. Si la subred a la que estas interfaces (i_m, j_n) fuera una interfaz punto a punto, se podría afirmar que la interfaz predecesora a j_n, la interfaz j_{n-1} forma un alias con la interfaz i_m, y del mismo modo, que la interfaz predecesora a i_m, la interfaz i_{m-1} forma un alias con la interfaz j_n, pues si las interfaces i_m, j_n formarían parte de un enlace punto a punto, solo tendrían dos nodos a los extremos del enlace. Sin embargo como se observa en la figura 2.5, solamente uno de los dos alias inferidos es correcto, solo el par i_{m-1}, j_n forman un alias.

La solución a este inconveniente y a otros problemas explicados en [GS09] que pueden ocasionar alias inferidos erróneamente, se solucionan cumpliendo determinadas condiciones antes de identificar un alias como verdadero, las que se explicarán a continuación.

2.2.3.3. Precisión en la identificación de Alias

i **Sin bucle**. Esta condición se basa en que los caminos descubiertos usando *traceroute* están libres de lazos y por tanto esta condición se debe mantener aun después de que dos o más interfaces sean consideradas como alias. Es decir, dadas dos interfaces $i_e^{v_p}, i_f^{v_p}$ candidatas a ser alias del *router* v_p, se considera este alias

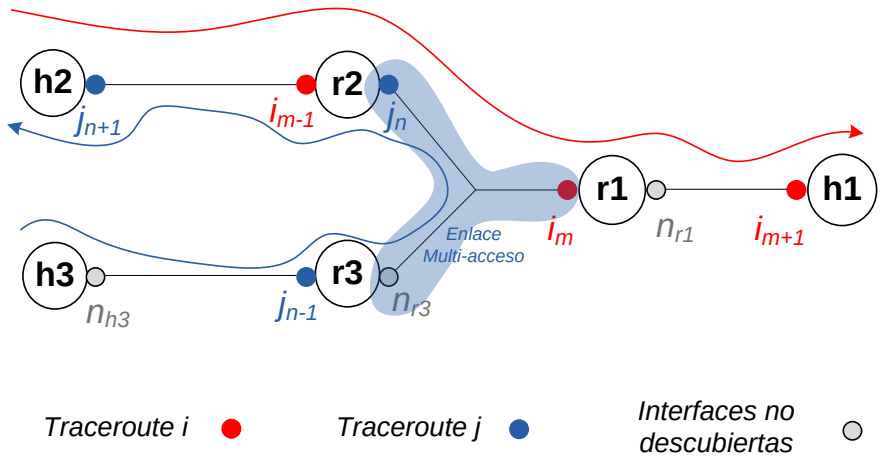


FIGURA 2.5: Resolución de Alias enlaces multi-acceso.

como verdadero solamente si la interfaz $i_e^{v_p}$ no aparece en ningún *traceroute* donde aparezca también la interfaz $i_f^{v_p}$.

ii **Vecinos en Común**. Esta condición establece que dadas dos interfaces candidatas a un alias, formarán un alias verdadero solamente si ambas interfaces tienen un mismo vecino en común que no sea alcanzado mediante alguna interfaz multi-acceso. Esta condición permite solucionar el inconveniente al inferir los enlaces multi-acceso explicado en la sección 2.2.3.2. Tomando como ejemplo nuevamente la Figura 2.5, cada par de interfaces i_{m-1}, j_n y i_m,j_{n-1} se consideran alias solamente si tienen un mismo vecino en común y si este vecino no es alcanzado a través de un enlace multi-acceso: Las interfaces i_{m-1}, j_n tienen como vecino en común la interfaz j_{n+1}, alcanzado a través de una interfaz punto a punto, y por tanto este par de interfaces es considerado como un alias; Las interfaces i_m, j_{n-1} por otro lado solo tienen vecinos en común a través de un enlace multi-acceso (interfaz j_n) y por tanto este alias será ignorado. Esta condición se aplica solamente cuando los alias se identifican a partir de enlaces que no son punto a punto.

La presencia de vecinos en común para un par de interfaces i_{m-1} y j_n, candidatas a alias descubiertas por los *traceroutes* i y j respectivamente, pueden detectarse de tres formas distintas:

- Las interfaces candidatas a alias i_{m-1} y j_n tienen una interfaz en común como vecina en algún *traceroute*,

ii Existe un alias previamente inferido (o,b) tal que b sea sucesor (o predecesor) de i_{m-1} y o sea predecesor (o sucesor) de j_n. Es decir, i_{m-1} y j_n no tienen interfaces como vecinas en común que se detecten de forma directa, pero tienen como vecino una interfaz perteneciente a un mismo alias previamente inferido o

iii La interfaz i_{m-1} tiene como sucesora una interfaz que pertenece a la misma subred de j_n, y la interfaz j_n tiene como sucesora una interfaz que pertenece a la misma subred i_{m-1}.

2.2.3.4. Reglas para la formación de subredes

Previamente se ha mencionado cómo funciona *APAR* para identificar los alias y las condiciones para que los alias sean inferidos con mayor precisión. *APAR* básicamente trata de alinear los *traceroutes* basándose en que los nodos pertenecientes a una misma subred forman un enlace entre ellos. Para aplicar esta premisa se ha considerado hasta el momento que ya se conocían las subredes de antemano, pero inicialmente esta información es desconocida, pues los únicos datos de entrada para el algoritmo son los *traceroutes*. Por tanto, esta sección explica el proceso para inferir las subredes a partir de la información obtenida de los *traceroutes*.

Primero se crean todas las subredes posibles para todas las *IP* que se hayan descubierto empezando por la máscara /22 hasta la máscara /31. Es decir, se comienza creando subredes con los 22 primeros bits del conjunto de direcciones *IP*. A continuación se procede a repetir este proceso con el resto de máscaras de subred (ejemplo: /23, /24, ..., /31).

Una vez creadas todas las subredes posibles se debe decidir cual es la subred que se aproxima más al valor real, por ejemplo: un conjunto de direcciones *IP* pueden formar parte de una subred mascara /29 o formar parte de dos subredes independientes de máscara /30. Esta discriminación para elegir la subred más adecuada se basa en las siguientes reglas:

i **Precisión**. Dos o más direcciones *IP* de una misma subred no pueden aparecer consecutivamente en un traceroute, si esto sucede esta subred será descartada.

ii **Completitud**. Si en los *traceroutes* no se encuentran utilizadas al menos la mitad de las direcciones *IP* disponibles de una determinada subred, esta subred será descartada. Una subred s de mascara x puede incluir hasta $2^{32-x} - 2$ *IP*. La relación entre direcciones *IP* encontradas en una subred y el número máximo de direcciones *IP* permitidas se conoce como factor de completitud

iii **Orden de procesamiento**. Se procesarán primero aquellas subredes con mayor factor de completitud, si este valor coincide para dos subredes entonces se procesará primero aquellas que aparezcan en la mayor cantidad de *traceroutes*. De esta forma, por definición las subredes que serán procesadas primero son todas las que posean máscara /30 y máscara /31.

De acuerdo a los autores, siguiendo las reglas previamente mencionadas, *APAR* proporciona una precisión del 95 % [GS09]. Para calcular la precisión, los autores calculan los errores sobre la red Abelin, una red de topología conocida. Por otro lado también usan el servicio de DNS para calcular los errores en base a los nombres que se resuelven de dos interfaces *IPs* que forman un alias, por ejemplo: Si una interfaz IP_1 se resuelve al nombre *ProveedorA.routerY.interfaz1* y una interfaz IP_2 se resuelve al nombre *ProveedorA.routerY.interfaz2*, y estas interfaces se han resuelto por *APAR* como un alias, entoces los autores consideran que el alias se ha inferido correctamente. Finalmente hacen comparaciones del algoritmo *APAR* con otros algoritmos de resolución de alias y consideran que el número máximo de alias que no coinciden sería el máximo error cometido por el algoritmo *APAR*, suponiendo que los otros algortimos son confiables.

2.2.4. Túneles *MPLS*

Multiprotocol Label Switching (MPLS) [RVC01] es un protocolo que fue inicialmente desarrollado para reducir el tiempo requerido en el envío de paquetes. Actualmente se despliega como solución para redes virtuales *VPN* [MM00] e ingeniería de tráfico [SVN04].

La arquitectura *MPLS* comienza en un *router IP* que agrega una o más etiquetas a cada uno de los paquetes recibidos. Esta etiqueta de 32 bits que se agrega antes del encabezado *IP* se denomina *label stack entry (LSE)* y determina las decisiones que tomarán los siguientes *routers* capaces de conmutar etiquetas o *Label Switching Router*

(*LSRs*) en la red. Los caminos formados de las conexiones de varios *LSRs* forman los distintos *Label Switching Path (LSP)* que pueden seguir los paquetes.

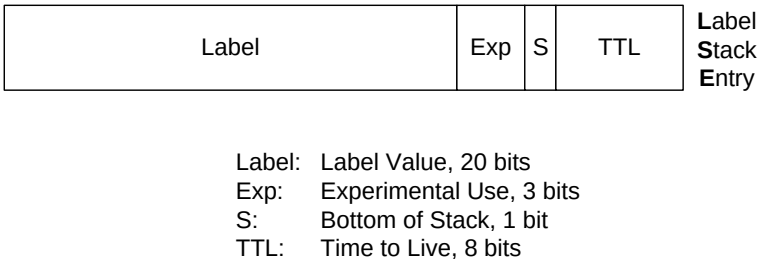


FIGURA 2.6: *MPLS label Stack.*

En una red *MPLS* los paquetes son enviados analizando la etiqueta de 20 bits del *LSE*. El *LSE* también tiene un campo *time-to-live* TTL_{lse} (ver Figura 2.6). En cada salto *MPLS* la etiqueta del paquete de entrada es reemplazada por la correspondiente etiqueta de salida. El *fowarding MPLS* es más ligero y rápido que el *fowarding IP* debido a que es más simple encontrar la coincidencia para una etiqueta *MPLS* fija, en lugar que para un prefijo *IP*, en donde es necesario encontrar también la máscara de red adecuada analizando todas las posibles subredes a partir de un prefijo dado.

Los *routers MPLS* tienen la opción de enviar mensajes *ICMP* `time-exceeded` cuando el TTL_{lse} del paquete expira como lo hacen los *routers IP*. También pueden implementar la RFC 4950 [BGTP07], una extensión de *ICMP* que permite al *router* añadir los campos del *LSE* en el mensaje *ICMP* `time-exceeded`. Es decir, básicamente copian los campos *MPLS* del paquete que ha expirado en el mensaje *ICMP* `time-exceeded`. Actualmente varias versiones de *traceroute* para *linux*[11], así como la versión actual de *paris-traceroute*[12] permiten visualizar los valores de la etiqueta *MPLS* junto al *TTL* que ya se mostraba tradicionalmente a la salida del *traceroute*, indicando así de manera explícita los *LSRs* atravesados por el *traceroute*. Aún cuando la *RFC 4950* no esté implementada, los *LSRs* que forman un túnel *MPLS* pueden detectarse si el primer *router* del *LSP* (el *Label Edge Router - LER* de ingreso) copia el valor del TTL_{ip} al campo TTL_{lse}, opción denominada `ttl-propagate`. Con lo cual cuando expire el TTL_{lse}, cada *LSR* se anunciará normalmente como un *router IP* mediante mensajes *ICMP*. Si el TTL_{lse}

[11] traceroute-nanog ftp://ftp.login.com/pub/software/traceroute/
[12] http://code.google.com/p/paris-traceroute/

no vence, su valor se copia nuevamente al campo TTL_{ip} cuando la etiqueta *MPLS* es removida por el *Label Edge Router - LER* de salida.

En ambas situaciones descritas los *LSRs* son visibles al *traceroute*, si la *RFC 4950* está implementada se usará la información adicional del mensaje *ICMP* para identificar los *routers* que forman parte del túnel *MPLS*, mientras que si solamente está implementada la opción `ttl-propagate` los *LSRs* serán visibles al *traceroute* pero no podrán identificarse como parte de un túnel *MPLS*

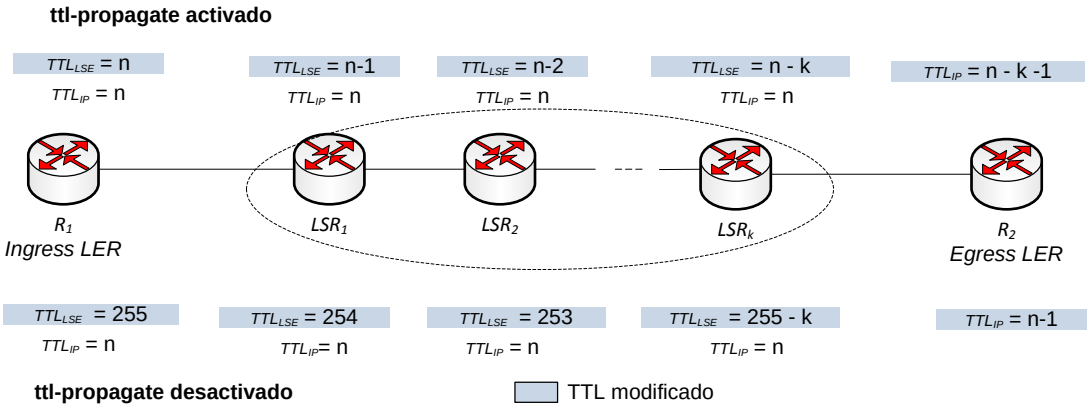


FIGURA 2.7: Propagación del *TTL* dentro del túnel *MPLS* cuando se encuentra activada y desactivada la opción `ttl-propagate`.

Desafortunadamente, no todos los *routers* están configurados con las opciones `ttl-propagate` o implementan la *RFC 4950*. Si la opción `ttl-propagate` no está habilitada el valor TTL_{ip} no se copia al campo TTL_{lse} y en lugar de ello éste se inicializa con un valor aleatorio generalmente cercano a potencia de 2: 32, 64, 128 y 255 [VPMD13]. En este caso el TTL_{lse} no tiene relación alguna con el valor del TTL_{ip}, el cual se disminuirá en uno solamente al salir del túnel *MPLS*. Bajo estas circunstancias el TTL_{ip} nunca expira dentro dentro del túnel *MPLS* por lo que los *LSRs* no pueden descubrirse por medio del *traceroute* [SBE11]. La Figura 2.7 muestra los dos casos de propagación del *TTL* dentro de un túnel *MPLS*.

Basado en lo expuesto previamente, *Donnet et al.* [DLMP12] propone una taxonomía basada en cuatro clases de túneles *MPLS*, mostrados en la figura 2.8 y descritos en las siguientes subsecciones.

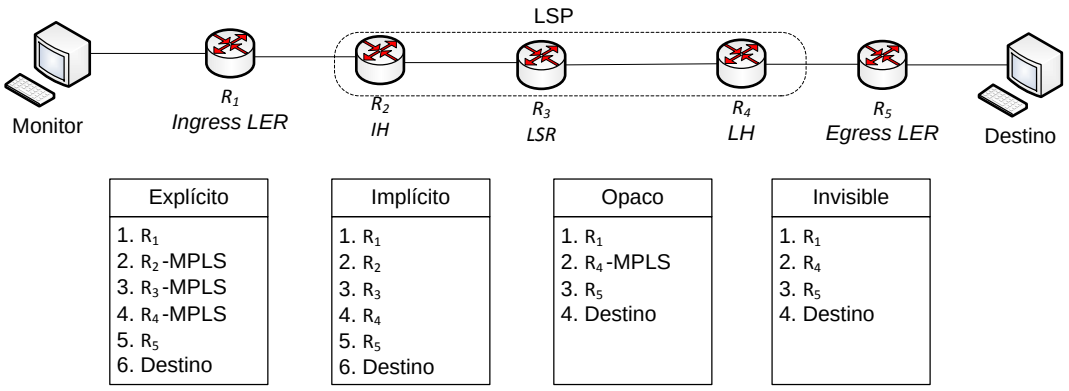


FIGURA 2.8: Taxonomía de los túneles *MPLS*.

2.2.4.1. Túneles explícitos

Los túneles explícitos se identifican mediante aquellos *LSRs* que tienen habilitadas tanto la opción `ttl-propagate` como la *RFC 4950*. Esto permite que el valor con el cual el TTL_{lse} se inicializa, corresponda con el valor que el campo TTL_{ip} tiene al ingresar al túnel *MPLS*, haciendo que todos los *LSRs* sean visibles como si fueran *routers IP* tradicionales. Por otro lado, los *LSRs* indican de manera explícita en el mensaje *ICMP* `time-exceeded` que son parte de un túnel *MPLS* (*4950*). Se ha encontrado que al rededor del 30 % del total de los caminos descubiertos en base a *traceroutes* tienen al menos un túnel *MPLS* explícito [SBE11] [DLMP12].

2.2.4.2. Túneles Implícitos

En este caso, los *LSRs* implementan la opción `ttl-propagate` pero no implementan la *RFC 4950*. Esto hace que los *LSRs* sean visibles al *traceroute* pero en primera instancia se desconozca si un *router* forma parte o no de un túnel *MPLS*. En [DLMP12] se proponen dos técnicas para inferir si un *router* pertenece o no a un túnel *MPLS*, aún cuando esta información no esté disponible de manera explícita, la primera técnica se basa en lo que los autores denominan *quoted-TTL* y la segunda en lo que denominan *u-turn signature*.

i ***Quoted-TTL* (q-ttl)** Esta firma se basa en que el valor TTL_{lse} es el que expira dentro del túnel *MPLS* y no el TTL_{ip}, el cual no varía a partir del momento en que el paquete *IP* ingresa al túnel *MPLS*. Por tanto si se recibe un mensaje

ICMP `time-exceeded` a partir de un paquete en el cual $TTL_{ip} > 1$ se puede asegurar que el *router* que generó dicho mensaje estaba dentro de un túnel *MPLS*, pues la única posibilidad es que dicho mensaje se haya generado debido a que expiró el campo TTL_{lse}. Esta información puede ser obtenida analizando los datos del mensaje *ICMP* `time-exceeded` recibidos como respuesta al *traceroute* ya que en ellos se añade el encabezado *IP* del mensaje original en donde se incluye el TTL_{ip} [Pos81]. A este valor del TTL_{ip} extraído del campo de datos del mensaje *ICMP* `time-exceeded` se le llama *quoted-ttl* y siempre que *quoted-ttl*> 1 existe un túnel *MPLS*.

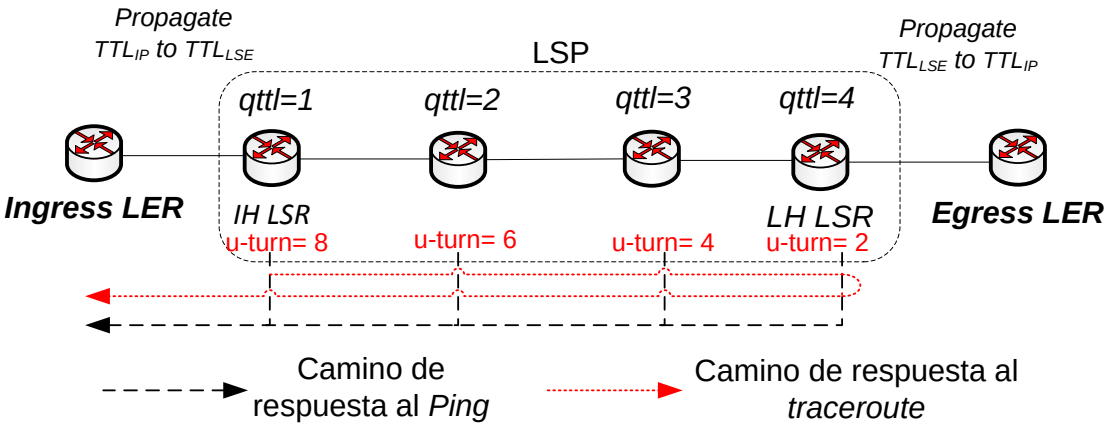


FIGURA 2.9: Túneles *MPLS* implícitos identificados por el *q-ttl* y por la firma *u-turn*.

El valor *quoted-ttl* permite también conocer la posición del *LSRs* dentro del túnel como se ilustra en la 2.9

ii ***U-turn signature*** Esta firma se basa en el hecho de que para ciertos fabricantes de *routers*, las configuraciones por defecto permiten que los *LSRs* en un túnel *MPLS* presenten un comportamiento en común: cuando el TTL_{lse} expira el *ICMP* `time-exceeded` generado se envía primero al último *hop* (*Last Hop (LH)*) del *LSP* y desde ahí se reenvía la respuesta hacia *host* que originó el paquete. Sin embargo, para el resto de paquetes como *ICMP* `echo`, los *LSRs* son capaces de responder directamente hacia el *host* origen. Esta diferenciación en el comportamiento entre los mensajes `echo-reply` y `time-exceeded` permite que se puedan identificar los *LSRs* de un túnel *MPLS*.

El método consiste en enviar un *traceroute* que genera un *ICMP* `time-exceeded` en cada uno de los *hops*, seguido de un *ICMP* `echo` hacia cada *hop* descubierto por el

traceroute. En el *host* origen se comparan los resultados recibidos, las diferencias en el TTL de los mensajes determinan la existencia del túnel MPLS. Es decir, si $TTL_{echo-replay} - TTL_{time-exceeded} > 0$ se dice que existe un túnel *MPLS* y además el valor de la diferencia del TTL varía a lo largo del *LSP* de la forma $X, X-2, X-4, X-6, ..., 2, 0$, donde X corresponde a dos veces la longitud del túnel, como se muestra en la Figura 2.9.

Se ha encontrado que al rededor del 5.5 % del total de direcciones *IP* que pertenecen a un *LSR* se descubren solamente a través de *u-turn signature* o *quoted-ttl* y que los túneles implícitos son 3 veces menos frecuentes que los túneles explícitos [DLMP12].

2.2.4.3. Túneles Opacos

Son aquellos en donde el *LER* de ingreso no tiene habilitada la opción `ttl-propagate` pero el último *router* del *LSP* o *LH* tiene implementada la *RFC 4950*. Cuando la opción `ttl-propagate` no está activada, el *LER* de ingreso inicializa el TTL_{lse} en 255 haciendo que el paquete nunca expire en el túnel y por tanto todos los *LSRs* se oculten al *traceroute*, excepto el *LH* que removiendo la etiqueta *MPLS* vuelve a decrementar el TTL_{ip} en lugar del TTL_{lse} como lo hicieron sus predecesores (ver figura 2.8).

Cuando el *LH* recibe el paquete aún con la etiqueta *MPLS* le corresponde un $TTL_{lse} = 255-(n+1)$, para un túnel de n *hops*. En [DLMP12] se menciona que si está habilitada la *RFC 4950*, los *routers* forman el mensaje `time-exceeded` incluyendo la etiqueta *MPLS* en él, con lo cual se puede determinar en base al TTL_{lse} ($1 < TTL_{lse} < 255$) añadido en el `time-exceeded`, la existencia de un túnel opaco y su longitud.

Estudios posteriores, encontraron que la frecuencia de aparición de este tipo de túneles es considerablemente baja (alrededor del 1 % de los túneles MPLS) y que la taxonomía descrita para identificarlos corresponde más una rara excepción de la regla, en donde la regla la forman los túneles invisibles [VPMD13].

2.2.4.4. Túneles Invisibles

Los túneles invisibles no implementan ni la opción `ttl-propagate` ni la *RFC 4950*, haciendo que hasta el momento no exista un estimado de su presencia en Internet.

2.3. Métricas para evaluar la Topología de Internet

2.3.1. Distribución de Grados

Representa la probabilidad de que un vértice tenga grado g. Una de las formas de expresarla es:

$$P(g) = \frac{1}{n} \sum_{\forall i / g(i)=g} 1 \tag{2.1}$$

donde $g(i)$ es el grado del vértice i, y n es el número total de vértices. Varios estudios han revelado que los grafos muestran una probabilidad $P(g)$ con cola pesada, que puede aproximarse a una ley de potencias, esto es

$$P(g) \approx g^{-\gamma}, \tag{2.2}$$

donde $2 \leq \gamma \leq 3$ [HFc12].

2.3.2. Distribución del grado medio de los vecinos

Representa el grado medio de todos los vecinos a un vértice de grado g, se define como:

$$g_{nn}(g) = \frac{1}{n_g} \sum_{\forall j / g_j=g} \frac{1}{|V(j)|} \sum_{i \in V(j)} g_i \tag{2.3}$$

donde $V(j)$ es el conjunto de los vecinos del vértice j, $|V(j)|$ su cardinalidad, g_i es el grado del vértice i, y n_g es el número de vértices de grado g. En el caso que los vértices de grado elevado posean un g_{nn} pequeño, entonces ellos actuarán de concentradores, implicando que los vértices de grado bajo tengan como vecinos otro de alto grado. Este comportamiento está definido como discordante [New02] y se observa en la topología de *AS*. En cambio cuando los vértices de grado elevado tengan como vecinos otros de grado también elevado, se denomina concordante [New02]. Este último comportamiento es observado en algunas redes sociales.

Las definiciones de discordante o concordante tienen sentido cuando las pendientes son marcadas, si no las pequeñas variaciones pueden ser originadas por los sesgos en la obtención de los mapas [AH06].

2.3.3. Distribución del coeficiente de *clustering*

El coeficiente de *clustering* indica qué probabilidad tienen los vecinos de un vértice de estar interconectados entre sí:

$$cc_i = \frac{2n_{link}}{g_i(g_i - 1)} \tag{2.4}$$

donde n_{link} representa el número de conexiones o aristas entre los vecinos de i, y g_i es el grado del vértice i. Su distribución en función del grado es:

$$c(g) = \frac{1}{n_g} \sum_{j/g_j=g} cc_j \tag{2.5}$$

Tanto $c(g)$ como $g_{nn}(g)$ están estrechamente relacionados ya que para tener un elevado coeficiente de *clustering* es necesario que el grado de los vecinos sea elevado. Otra observación útil es que estos dos parámetros permiten diferenciar si el grafo representa a una topología de Internet a nivel AS o a nivel *routers*. En el primer caso el comportamiento es discordante para ambas curvas con una pendiente negativa, mientras que para la topología a nivel de *routers* ambas tienen un comportamiento casi estático con una pendiente tendiendo a cero [PSV04, pág 21].

2.3.4. Visualización del Grafo mediante *k*-núcleos

Una manera relativamente nueva para caracterizar y analizar distintos tipos de redes es mediante la visualización del grafo basándose en la descomposición en k-núcleos [AHBV06].

Considerando el grafo $G = (V, E)$, la descomposición en k-núcleos se basa en la siguientes definiciones:

i ***k*-núcleos**. Un subgrafo $H = (C, E|C)$ inducido por el conjunto de vértices $C \subseteq V$ es un k-núcleo o núcleo de orden k *sii* $\forall v \in C : grado_H(v) \geq k$ y H es el máximo subgrafo con esta propiedad.

Una forma de obtener la descomposición en k-núcleos es eliminando recursivamente todos los vértices de grado menor que k, hasta que todos los vértices restantes tengan grado mayor o igual a k.

ii **Capa**. Un vértice i tiene número de capa c si dicho vértice pertenece al c-núcleo pero no al $(c+1)$-núcleo. La capa a la que pertenece el vértice i se denomina c_i

iii **Cliqué**. Un cliqué es un subgrafo donde cada vértice está conectado a cada uno de los vértices del grafo. Esto equivale a decir que el subgrafo inducido por V es un grafo completo.

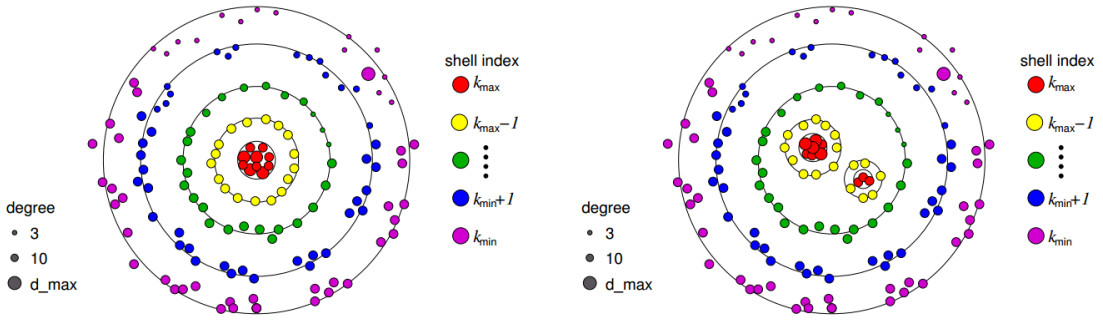


FIGURA 2.10: Visualización basada en k-núcleos. A la izquierda todos los k-núcleos son conexos; a la derecha algunos k-núcleos están conformados por más de una componente conexa [AHBV06].

La visualización basada en k-núcleos básicamente consiste en un gráfico de dos dimensiones compuesto por una serie de capas circulares y concéntricas (ver Figura 2.10). Cada círculo corresponde a cada capa k y todos los vértices que la conforman se dibujan sobre ella con el mismo color. El círculo exterior corresponde a la capa k_{min}-núcleo mientras que el círculo interior corresponde a la capa k_{max}-núcleo. El diámetro con el que se dibuja cada capa depende del valor de k y es proporcional a $k_{max} - k$. De esta forma cada capa se añade de forma ordenada. Por otro lado, cuando el k-núcleo está fragmentado en dos o más componentes, el diámetro de cada capa depende además del número de vértices que pertenecen a ella. Finalmente, el tamaño de cada nodo es proporcional logarítmicamente al grado original de dicho vértice [AHBV06].

Actualmente, esta visualización puede implementarse mediante la herramienta *open-source LaNet-vi*[13], proyecto llevado a cabo entre la Universidad de Buenos Aires, Indiana

[13] http://lanet-vi.fi.uba.ar/

University, CNRS y Politécnico di Torino. Una versión *on-line* de la herramienta se encuentra también disponible.

Capítulo 3

Experiencias y análisis de las exploraciones

La presente sección detalla las características del experimento realizado y el análisis de los resultados obtenidos. Inicialmente se describe brevemente la plataforma de *Planet-lab*, a continuación se explica el proceso de recolección de datos con todas las variables y herramientas involucradas, donde se explica entre otras cosas el tamaño de la exploración, y las características de la herramienta de exploración. Posteriormente, se describe el método seguido para la identificación de los nodos *MPLS* y se discuten los resultados de este proceso.

Más adelante, se explica cómo se forman las distintas topologías que permitirán analizar el comportamiento de los enlaces *MPLS* inferidos y se presentan los resultados.

Finalmente, se discute el comportamiento de los enlaces *MPLS* dentro y fuera de un propio *AS*.

3.1. *Planet-lab*

Los datos se obtuvieron utilizando la plataforma de *Planet-lab*[1]. Esta es una plataforma constituida por un gran número de servidores distribuidos a través de varias redes

[1] https://www.planet-lab.org/

académicas formando un laboratorio a escala mundial. Actualmente cuenta con más de 1000 nodos distribuidos en cerca de 500 sitios distintos.

Planet-lab permite acceder a uno o varios espacios virtuales en cada uno sus nodos, en donde se puede desarrollar, instalar y ejecutar aplicaciones. Los servicios y aplicaciones de *Planet-lab* a los que se puede acceder como usuario, corren sobre una máquina virtual instalada en distintos nodos que comparten una fracción de sus recursos de hardware. Si bien el concepto de máquina virtual está ampliamente desarrollado, *Planet-lab* introduce el concepto de virtualización distribuida, que permite tratar varias máquinas virtuales como si fueran una solo entidad. Cada nodo de *Planet-lab* posee las siguientes características mínimas:

i ***Harwdware* mínimo**. El *hardware* mínimo solicitado por *planet-lab* ha ido variando con el tiempo. Actualmente cada nodo debe tener al menos 24 *GByte* de memoria RAM, un CPU de similares características a un 6x Intel Xeon E5 cores @ 2.2Ghz y 2TB de almacenamiento. Posteriormente estos recursos son compartidos virtualmente.

ii **Requerimientos mínimos de Red**.*Planet-lab* solicita una conexión a Internet de al menos 400 *kbps* de ancho de banda simétrico. Este valor no ha sido actualizado por la comunidad de *planet-lab* desde el 2007, de modo que es de esperarse que normalmente las velocidades de conexión sean en la actualidad mucho mayores.

Entre las limitaciones de *planet-lab* se encuentra la competencia por los recursos de *hardware* con otros usuarios de la virtualización distribuida, con lo cual existe la posibilidad de no conseguir las mismas condiciones de operación a lo largo de las pruebas. Sin embargo, esta limitación no es significativa para los experimentos del presente trabajo, ya que las pruebas no fueron exigentes con el *hardware* requerido. Se desarrolló una herramiento de exploración bastante simple que se analiza más adelante.

3.2. Recolección de Datos

Se realizaron varias exploraciones de prueba de distintas características variando el tamaño de la exploración y el número de repeticiones por cada exploración. El tamaño de

la exploración se relaciona con la cantidad de nodos origen y nodos destino; el número de repeticiones es el número de exploraciones realizadas entre un mismo par origen - destino. El motivo de estas pruebas fue afinar los parámetros para realizar la exploración definitiva, teniendo por objeto encontrar la menor cantidad de información redundante y a la vez confiable. Se encontró que no resulta eficiente aumentar el número de repeticiones por exploración, puesto que los resultados obtenidos no cambiaron significativamente de una exploración a otra, por otro lado, al aumentar el tamaño de la exploración se obtiene más información útil, es decir, se descubren más enlaces. Para llegar a esta conclusión se analizó si los resultados de los grafos obtenidos cambiaban cuando se repetía la exploración varias veces por día o por semana, conservando los mismos nodos de origen y los mismos nodos de destino. Se encontró que los resultados no cambian significativamente al repetir varias veces un mismo experimento:los grafos obtenidos con una y varias repeticiones tenían tamaños similares entre sí y mantenian las propiedades en la distribución de grado, en el grado medio de los vecinos, coeficiente de *clustering* y composición *k-cores*. Es por esto que en la exploración definitiva se dio prioridad al tamaño de la exploración y no al número de repeticiones.

El presente trabajo utilizó varios nodos de *Planet-lab* en los cuales se alojó la herramienta de exploración basada en *paris-traceroute* para la recolección de resultados. Los datos se recolectaron entre el 7 y el 10 de Noviembre del 2013. A continuación se detalla el trabajo realizado al respecto.

3.2.1. Tamaño de la Exploración

Se usaron 434 nodos de *Planet-lab* como puntos de recolección de datos. Los nodos se eligieron al azar tratando en lo posible de evitar que más de dos nodos se ubiquen en el mismo sitio. Un sitio se define como la ubicación física donde el nodo de *Planet-lab* está ubicado (ejemplo: Universidad de Buenos Aires). Un nodo se define como un servidor dedicado que corre los servicios de *Planet-lab*.

Desde cada nodo seleccionado se realizó una exploración basada en *paris-traceroute* hacia 13153 destinos, entre los cuales se incluyen también los nodos de *Planet-lab* previamente elegidos como puntos de recolección de datos, el resto de los destinos se eligieron al azar de todo el rango de IPv4 públicas disponibles. Como resultado se enviaron un total de 4179924 *paris-traceroutes* de exploración.

Nodos de recolección de datos	**Nodos destino**	**Total de Exploraciones**
434	13153	4179924

TABLA 3.1: Tamaño de la Exploración.

3.2.2. Herramienta de Exploración

El tamaño de la exploración se definió tratando de maximizar la información que pudo ser procesada con el *hardware* y software disponible [2].

Se usó una herramienta basada en *paris-traceroute*, una variante del *traceroute* tradicional que evita que se descubran falsos enlaces (ver sección 2.2.1).

La herramienta se implementó en un *script* de *python* que se ejecuta desde cada uno de los nodos de recolección de datos, almacena los resultados en un texto plano que luego es descargado y almacenado en una base de datos para su posterior análisis.

La herramienta de exploración diseñada se puede explicar en dos módulos:

i El primer módulo funciona como *traceroute*, es decir descubre los nodos involucrados hasta llegar a la IP destino.

ii El segundo módulo funciona como *ping* hacia cada una de las *IPs* descubiertas previamente por el módulo *traceroute*.

Por tanto, en el primer módulo de la herramienta se obtiene como respuesta mensajes *ICMP* `time-exceeded` cuando expira el *TTL* del *traceroute*, mientras en el segundo módulo que actúa como *ping* se reciben mensajes *ICMP* `echo-reply` desde cada uno de los nodos descubiertos previamente. En ambas etapas cada *traceroute* o *ping* es basado en *paris-traceroute*.

Las características de la herramienta, que no es más que un *paris-traceroute* con distintas configuraciones, se muestran en la tabla 3.2. Esta herramienta permite crear los mensajes *ICMP* `time-exceeded` e *ICMP* `echo-reply` en cada uno de los nodos descubiertos, información que será usada previamente para analizar el valor de la firma `u-turn` (ver sección 2.2.4.2).

[2] Se utilizó una PC Intel Core i7 @ 1.73 GHz y 4 *Gbyte* de RAM

Descripción	**Usado como *traceroute***	**Usado como *ping***
Forma en la que se incrementará el *TTL*	*hop by hop*	*hop by hop*
Pruebas enviadas con el mismo *TTL* por salto	3	6
Tiempo que se esperará antes de marcar el mensaje como "sin respuesta"	1000 ms	1000 ms
Protocolo	*ICMP*	*ICMP*
TTL inicial a partir del cual arrancan las pruebas*	1	30
Número de mensajes "sin respuesta" consecutivos que finalizan el algoritmo**	4	4
IP Destino	Nodos de planetlab e *IPs* elegidas al azar	Cada una de las *IPs* descubiertas en el modo *traceroute*

TABLA 3.2: Parámetros del *paris-traceroute* usados en la herramienta de exploración.
* Cuando el *TTL* se inicializa en 1 se simula el funcionamiento del *traceroute* tradicional: Los paquetes expiran en el primer salto, el *TTL* se incrementa en uno, los paquetes expiran en el segundo salto y sucesivamente hasta llegar al destino. Cuando el *TTL* se inicializa en 30 los paquetes enviados no expiran, con lo cual se alcanza el destino, que usualmente se encuentra a menos de 30 saltos de distancia, simulando un *ping*.
** El algoritmo finaliza cuando se alcancen 4 mensajes sin respuesta consecutivos, ya sea que se originen en el mismo salto, o en dos saltos seguidos.

Se observó que *paris-traceroute* genera el mismo *checksum* para todos los mensajes *ICMP echo-request* aunque se envíen a destinos distintos. Esto permite afirmar que todos los mensajes enviados tanto en el módulo *traceroute* y módulo *ping* de la herramienta de exploración, en dirección origen a destino siguieron un mismo camino y no se vieron afectados por los balanceadores de carga por flujo; pues el balanceo de carga depende de que se mantengan constantes los campos: *checksum*, *code* y *type* (ver sección 2.2.1).

El hecho de que todos los mensajes de exploración enviados hacia un destino sigan un mismo camino evita detectar falsos enlaces, sin embargo en este punto se encontró una limitación: *paris-traceroute* permitió garantizar que solamente los mensajes de ida sigan un mismo camino, es decir no es bidireccional, pues a los mensajes correspondientes al módulo *traceroute* de la herramienta de exploración, se responde con un mensaje `time-exceeded` (*Code 11*), mientras que a los mensajes del módulo *ping* se responde con un `echo-reply` (*Code 0*). La variación de los valores del campo *code* del mensaje *ICMP* hace que sea imposible asegurar que los mensajes de vuelta sigan un mismo trayecto, pues solamente asegurando que todos los mensajes tengan el mismo *checksum*,

code y *type* se puede asegurar que serán parte de un mismo flujo y no se verán afectados por balanceadores de carga por flujo.

Está limitación nos advierte que ante la imposibilidad de que los mensajes *ICMP* `time-exceeded` y `echo-reply` sigan un mismo camino de vuelta, la firma *u-turn* resultará imprecisa (ver sección 2.2.4.2), pues considera que la variación entre el $TTL_{\texttt{time-exceeded}}$ y el $TTL_{\texttt{echo-reply}}$ se debe solamente a los túneles *MPLS* y no a que los mensajes han seguido rutas distintas. El impacto de esta limitación se analiza más adelante en la sección 3.3.4.3.

3.3. Identificación de túneles *MPLS*

En esta sección, en primer lugar se explica cómo se identificaron los túneles explícitos, implícitos y opacos (para entender mejor cada uno de ellos ver sección 2.2.4), usando herramientas como *tcpdump* y también las salidas del *paris-traceroute*.

Finalmente se presenta el análisis de los resultados obtenidos con respecto a los túneles *MPLS* encontrados.

3.3.1. Identificación de Túneles Explícitos

Estos túneles se identifican porque sus *LSRs* implementan la *RFC 4950* y la opción `ttl-propagate`. Si estas opciones están activas, *paris-traceroute* imprime la información de la etiqueta *MPLS* y el *q-tll* de cada nodo descubierto. Se utilizó esta información para identificar cuales de los nodos descubiertos forman parte de un túnel explícito. Se encontró que en el 39,6 % (ver figura 3.1) de los *paris-traceroutes* existen túneles *MPLS* explícitos.

3.3.2. Identificación de los Túneles Implícitos

Los túneles implícitos están compuestos por los *LSRs* que obedecen a las firmas *q-ttl* o *u-turn*. Se encontró que un total de 6,8 % de los *paris-traceroute* está compuesto por túneles implícitos.

i ***Routers* que obedecen a la firma q-ttl**. Son los *LSRs* que no implementan la *RFC 4950* pero tienen activa la opción *ttl-propagate*. Estos *LSRs* se identificaron utilizando un filtro en *tcpdump* que capturó todos los paquetes con *q-ttl* > 1 y sin la *RFC 4950*. Se encontró que un $4,7\,\%$ (ver figura 3.1) de los *paris-traceroutes* están compuestos por túneles implícitos que obedecen a la firma *q-ttl*.

ii ***Routers* que obedecen a la firma u-turn**. Estos *LSRs* se descubren evaluando las diferencias entre el $TTL_{\texttt{time-exceeded}}$ y el $TTL_{\texttt{echo-reply}}$ recibidos como consecuencia del *paris-traceroute* funcionando como *traceroute* tradicional y como *ping* (ver tabla 3.2). Además se aplicaron dos filtros sobre los paquetes recibidos: El primer filtro descartó todos los candidatos a *LSRs* que generaron distintos *TTL* para un mismo tipo de mensaje *ICMP*, es decir, para que no sean descartados los candidatos a *LSRs*, los 3 valores de $TTL_{\texttt{time-exceeded}}$ que registró un *LSRs* deben ser iguales entre sí y los 6 valores de $TTL_{\texttt{echo-reply}}$ también. Esto permite tener mayor confianza en que los mensajes recibidos hayan seguido una misma ruta y hayan evitado los balanceadores de carga, ya que no se puede garantizar nada sobre el camino de regreso aún usando *paris-traceroute*. El segundo filtro descarta a todos los candidatos a *LSRs* en donde *u-turn* ≤ 4, en lugar de *u-turn* ≤ 3 tal como se recomienda en [DLMP12] (esto se explicará en la sección 3.3.4.3). Es decir, solamente si $TTL^r_{\texttt{echo-reply}} - TTL^r_{\texttt{time-exceeded}} > 4$, se considerará que el *router* r es un nodo *MPLS*. Este filtro permite que las variaciones del *TTL*, ocasionadas por los caminos distintos que pueden tomar los mensajes *ICMP* `time-exceeded` y `echo-reply`, generen valores de *u-turn* > 4 y se infiera erróneamente un túnel *MPLS*. Por supuesto, garantizar que el valor de *u-turn* sea mayor a 4 no descarta que la diferencia $TTL^r_{\texttt{echo-reply}} - TTL^r_{\texttt{time-exceeded}}$ se produzca porque cada *ICMP* siguió caminos distintos, en vista de ello, más adelante (sección 3.3.4.3) se analizará la validez de esta firma. Se encontró que el $2{,}8\,\%$ de los *paris-traceroutes* está compuesto por túneles que obedecen a la firma *u-turn* (ver figura 3.1).

3.3.3. Identificación de los Túneles Opacos

Los túneles opacos contienen los *LSRs* que implementan la *RFC 4950* pero tienen desactivada la opción `ttl-propagate`. Para identificar estos *LSRs* se usó la herramienta *tcpdump* con un filtro que capture todos los paquetes en donde $1 < TTL_{lse} < 255$ y la

RFC 4950 no esté implementada. Al estar desactivada la opción de `ttl-propagate`, el paquete nunca va a expirar dentro del túnel *MPLS* y solo expirará en el último *LSR* del túnel, el cual analizará la etiqueta *IP*. Por otro lado al estar habilitada la *RFC 4950*, el mensaje `time-exceeded` adjuntará la información de la etiqueta *MPLS*, donde se incluye el TTL_{lse} que nunca expiró en el túnel y por tanto es mayor a 1.

No se encontró evidencia de la existencia de túneles que obedezcan a esta taxonomía. Esto podría resultar extraño, sin embargo como se menciona en la sección 2.2.4.3, los túneles opacos corresponden a una rara excepción a los túneles invisibles y no una configuración típica. Además de la baja frecuencia que sugieren los autores en la aparición de este tipo de túneles (20 veces menos que los túneles explícitos), la ausencia de túneles opacos en el presente trabajo podría significar que aún cuando está implementada la *RFC4950*, el mensaje *ICMP* `time-exceeded` es generado al analizar el paquete *IP* una vez retirada la etiqueta *MPLS* y por tanto, esta no se incluiría en el mensaje *ICMP*.

3.3.4. Análisis de los Datos

Se encontró que un total del $42,9\,\%$ de los *paris-traceroutes* está compuesto por túneles *MPLS*, el $39,6\,\%$ de los *paris-traceroutes* está compuesto por túneles *MPLS* explícitos y el $6,8\,\%$ está compuesto por túneles implícitos. Un $4,7\,\%$ de los *paris-traceroutes* está compuesto por túneles *MPLS* implícitos que obedecen a la firma *q-ttl* y un $2,8\,\%$ de los *paris-traceroutes* está compuesto por túneles implícitos que obedece a la firma *u-turn*. Finalmente, no se encontró evidencia de la existencia de túneles opacos que obedezcan a la taxonomía descrita en [DLMP12]. Los resultados obtenidos se resumen en la figura 3.1.

Más detalles sobre la distribución de los túneles *MPLS* encontrados se proporcionan la figura 3.2, de donde se observa que la cantidad de túneles *MPLS* atravesados por cada monitor es mayor al $30\,\%$ en cerca del $80\,\%$ de monitores, coincidiendo con los valores obtenidos previamente en [DLMP12] y [SBE11]. Esta observación muestra la importancia que en la actualidad tienen los túneles *MPLS* en Internet, pues su presencia prevalece sin importar la ubicación geográfica de los monitores. Los monitores donde casi no se observa presencia de túneles *MPLS* representan monitores que estuvieron la mayor parte del tiempo fuera de servicio y por tanto sus resultados no representan una muestra significativa. Por otro lado, aquellos monitores en donde casi el $99\,\%$ de

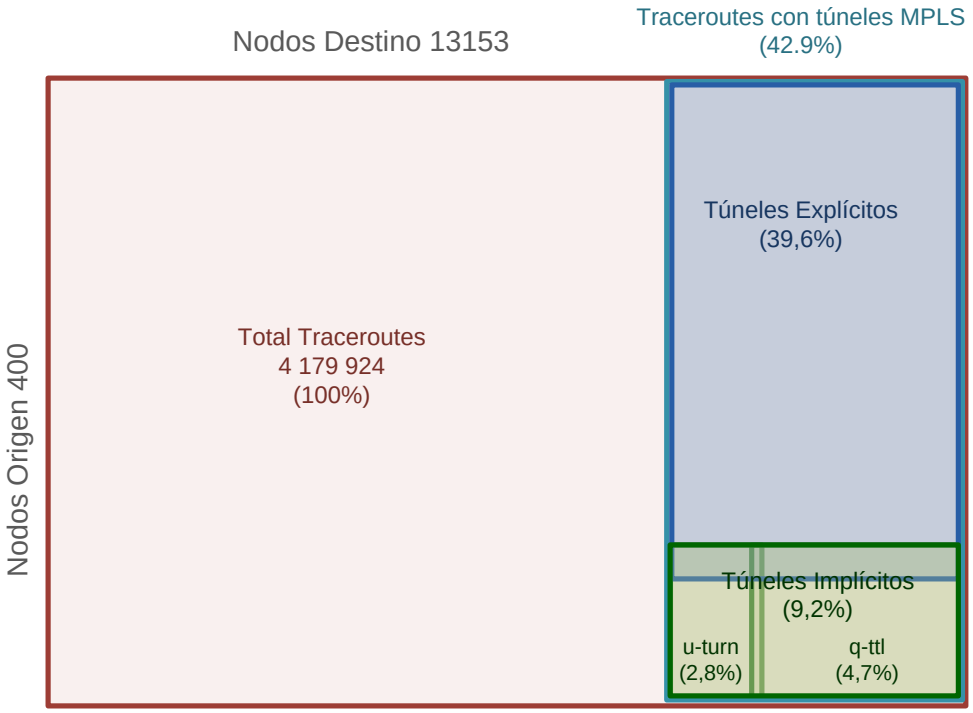


FIGURA 3.1: Resultados obtenidos en la identificación de túneles *MPLS*.

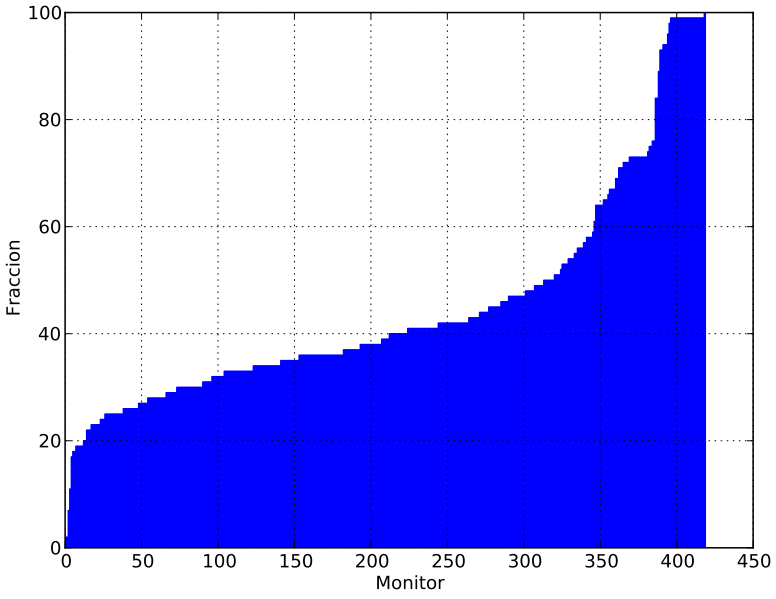


FIGURA 3.2: Fracción de *paris-traceroutes* que atraviesan un túnel *MPLS* por monitor, ordenados en forma ascendente acuerdo al número de túneles *MPLS* encontrados. El eje x representa cada uno de los monitores desde los cuales se realizaron las exploraciones y el eje y representa la fracción de *paris-traceroutes* en los cuales se encontró al menos un túnel *MPLS*.

paris-traceroutes atraviesan un túnel *MPLS*, son casos aislados en donde se cree que el monitor está ubicado en un *ISP* que usa directamente una red *MPLS*.

La distribución de la longitud de los túneles *MPLS* se muestra en la figura 3.3. La cantidad de túneles *MPLS* decae con la longitud de forma casi exponencial, predominando en más de la mitad de los túneles analizados los túneles con longitud 1 y 2. Por otro lado, se considera prudente considerar 9 a la longitud máxima alcanzada por los túneles. La longitud máxima está condicionada a los errores que pueden acarrearse del reconocimiento de falsos *routers MPLS*, principalmente debido a la firma *u-turn*, por tanto no se puede asegurar que la longitud máxima de un túnel *MPLS* sea 15 como se muestra en la figura. Del análisis de los resultados se obtuvo un $q\text{-}ttl_{max} = 9$, con lo cual se puede asegurar con certeza que existen túneles de hasta longitud 9. En los túneles de mayor longitud es imposible asegurar si hubo o no errores debido a la inexactitud de la firma *u-turn*. De todas formas, solamente el $1,78\,\%$ de los túneles reportan una longitud mayor a 9.

El hecho de que prevalezcan los túneles con longitud menor o igual a 2 significa que en más de la mitad de los casos, la red destino se alcanzó atravesando 1 o máximo 2 *LSRs* de una red *MPLS*, con lo cual se sospecha que estas redes tienen elevada conectividad internamente o bien tienden a ser de tamaño relativamente pequeño.

En las siguientes secciones se analiza la validez de las firmas *u-turn* y *q-ttl*. Para el análisis planteado se parte del hecho que el valor de ambas firmas tiene que ver con la posición que el *LSR* ocupa en el interior del túnel *MPLS*, para esto se encuentra la posición en la que cada uno de los *LSRs* fue detectado y se compara esta posición con los valores de las firmas *u-turn* y *q-ttl*.

3.3.4.1. Posición n dentro del túnel *MPLS*

Antes de analizar las firmas *q-ttl* o *u-turn*, en esta sección se define la posición n dentro de un túnel *MPLS*.

El valor de n representa la posición en la que un *LSR* es descubierto dentro del túnel *MPLS*. Este valor se obtiene observando el orden en el que un *LSR* ha sido descubierto: el valor de n se inicializa en 1 para el primer *LSR* descubierto y aumenta en 1 siempre

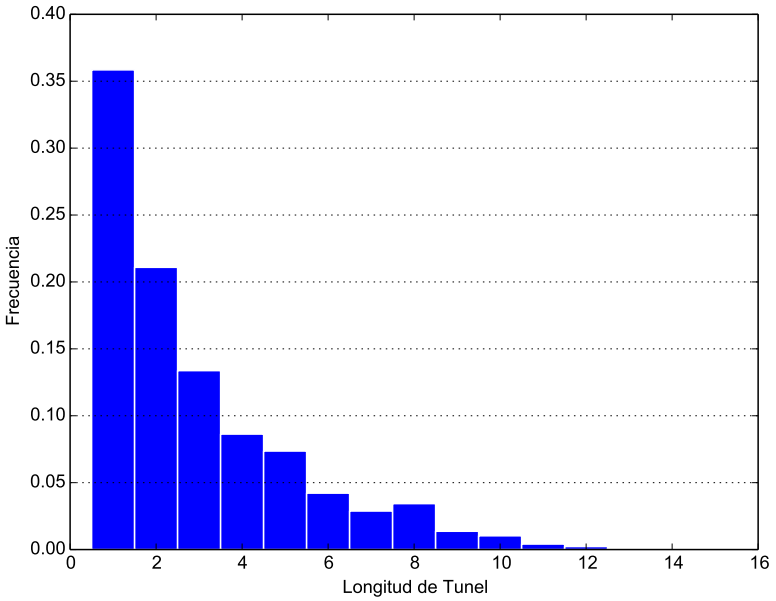


FIGURA 3.3: Histograma de la longitud de los túneles *MPLS*. Se considera una muestra de 1795158 túneles *MPLS*.

y cuando se detecte un *LSR* consecutivo. Este simple proceso se puede entender mejor observando el algoritmo 3.1.

Se debe notar que el valor de la posición n es independiente del valor numérico de las firmas *q-ttl* o *u-turn*, y depende solamente de si el *router* forma parte del túnel MPLS o no, y es precisamente por ello que se optó por comparar este valor con el valor anunciado por las firmas, para comparar la posición en el que un *LSR* aparece dentro del túnel con el valor que anuncia su firma.

Recordando la taxonomía de un túnel *MPLS*, se considera que el *router* es parte del túnel *MPLS* cuando se cumple cualquiera de las siguientes condiciones:

i el *router* implementa la *RFC 4950* (túnel explícito).

ii se detecta la firma *q-ttl* > 1 aún cuando no implementa la *RFC 4950* (túnel implícito).

iii se detecta la firma *u-turn* > 4 (túnel implícito).

Siempre y cuando se detecte cualquiera de las condiciones previamente planteadas, bajo las cuales un *router* se considera parte de un túnel *MPLS*, la posición n incrementará en

1 respecto a su valor anterior. Idealmente, resulta lógico creer que la posición n tendría que corresponderse con el valor de las firmas *q-ttl* o *u-turn*, sin embargo se hizo este análisis justamente para poner a prueba esta hipótesis.

El proceso mediante el cual se obtiene el valor de la posición n se resume en el algoritmo 3.1, en las líneas 4 y 5 se buscan las interfaces descubiertas mediante los *paris-traceroutes*, en la linea 6 se comprueba si la interfaz i_m analizada pertenece a un *router MPLS*, en las líneas 7 hasta 10, la posición n_m del *router MPLS* se actualiza incrementándose en uno, siempre y cuando la interfaz anterior i_{m-1} también haya pertenecido a un *router MPLS*, caso contrario se vuelve a fijar la posición n_m en 1, indicando que inicia un nuevo túnel *MPLS*.

Algoritmo 3.1 cálculo de la posición n.

Input: Traza del paris-traceroute con origen v_i y destino v_j donde se han identificado previamente los nodos *MPLS*: paris-traceroute(v_i, v_j)=$i_1, i_2^{mpls}, i_3^{mpls}..., i_j$

Output: Listado interfaces de los paris-traceroute(v_i, v_j) que pertenecen a túneles *MPLS* y su respectiva posición dentro del túnel: posición(v_i, v_j)= (i_2^{mpls}, n_m), (i_3^{mpls}, n_{m+1}),...

1: **for** $i_m \in$ paris-traceroute(v_i, v_j) **do**
2: **if** $i_m \in mpls$ **then**
3: **if** $i_{m-1} \in mpls$ **then**
4: $n_m = n_{m-1} + 1$
5: **else**
6: $n_m = 1$
7: **end if**
8: **append** (i_m, n_m) to posición(v_i, v_j)
9: **end if**
10: **end for**

3.3.4.2. Comportamiento de la firma *q-ttl* en el túnel *MPLS*

En la presente sección se analiza la validez de los resultados previamente obtenidos y el significado de la firma *q-ttl* en el túnel *MPLS*, para esto se ha usado el valor *q-ttl* (ver sección 3.3.4.2) y la posición n.

La Figura 3.4 corresponde a una gráfica de dispersión donde se muestra la correspondencia entre la firma *q-ttl* y la posición n que el *LSR* ocupa al interior del túnel *MPLS*. El diámetro de cada círculo es proporcional al número de *LSRs* que caen en cada posición n. Idealmente, la figura debería mostrar la perfecta correspondencia entre *q-ttl* y n, siguiendo la curva *q-ttl* $= n$. Sin embargo se observa que los valores de *q-ttl* se desvían

en $\Delta = \pm 1$, por ejemplo para $n = 5$ se tiene valores de *q-ttl* de 4 y 6. Por otro lado, se observa también que en todos los valores de n existe una considerable cantidad de firmas donde $q\text{-}ttl = 1$.

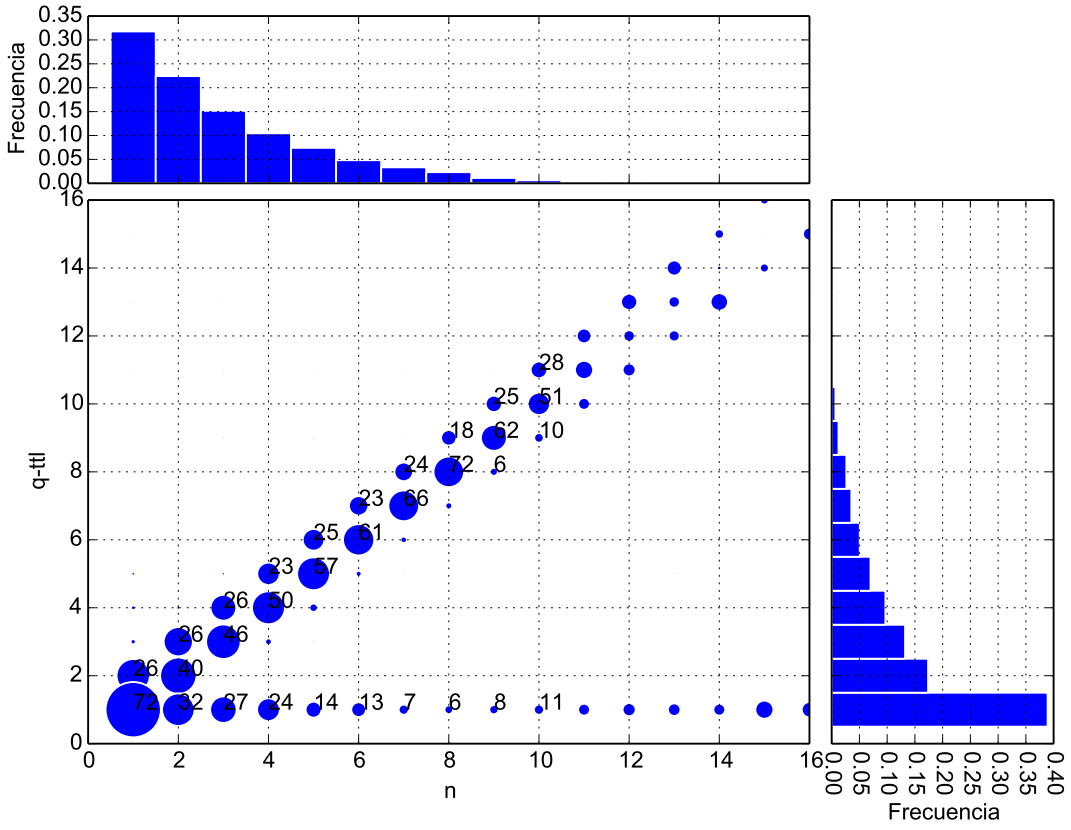


FIGURA 3.4: En la parte superior se muestra un histograma que representa la posición n de los *LSRs* dentro del túnel *MPLS*. A la derecha se muestra por otro lado, un histograma de los valores *q-ttl*. Finalmente el gráfico de dispersión compara los valores del *q-ttl* anunciados por los *LSRs* y su posición n. La etiqueta junto a cada círculo indica el porcentaje de *LSRs* que obedecen a cada valor de *q-ttl* para una misma posición n, por ejemplo: de los *LSRs* ubicados en la sexta posición, el 13 % tienen un *q-ttl* igual a 1, un 61 % tienen un *q-ttl* igual a 6 y un 23 % tienen un *q-ttl* igual a 7. Los valores menores al 10 % no se muestran en la gráfica por motivos estéticos. La figura representa una muestra de 6397056 *LSRs*.

Las variaciones $q\text{-}ttl = n \pm \Delta$ observadas en la figura 3.4, se deben a:

i ***q-ttl*** $= n + \Delta$: El valor de *q-ttl* por sobre la posición n se debe a un desfase al encontrar el primer *LSR*. Este desfase se explica considerando que el primer *LSR* del túnel (con $q\text{-}ttl = 1$), solamente puede ser detectado si implementa la *RFC4950*, de lo contrario se confunde como un *router IP* más, ya que el *q-ttl* expira en 1 en todos los *routers*. De este modo, si la *RFC4950* no está activa, un *LSR* tiene que tener un $q\text{-}ttl > 1$ para que sea considerado como parte de un túnel *MPLS*. De este

modo, un *LSR* con un valor de *q-ttl* = 1 sin la *RFC4950* será ignorado, el siguiente *LSR* con *q-ttl* = 2 será interpretado como el primer *LSR* del túnel, obtienen la posición un valor de $n = 1$, el segundo *LSR* dentro del túnel ($n = 2$) tendría un *q-ttl* = 3, etc., generándose un desfase ocasionado por la incapacidad de reconocer el primer *LSR* del túnel *MPLS*.

ii ***q-ttl*** $= n - \Delta$: El valor de *q-ttl* por debajo del valor esperado se debe a dos comportamientos que se cree se ocasionan a causa de políticas de balanceo de carga. Pues se encontró que aún cuando se usó *paris-traceroute* para mitigar el efecto producido por los balanceadores de carga, algunos mensajes pueden seguir caminos distintos debido a los balanceadores por *paquete*. Tomando como ejemplo la Figura 3.5 y considerando que el mensaje *ICMP* del *paris-traceroute* sale del origen con $TTL = 6$, siguiendo el camino $H_1, R_a, R_b, R_c, R_d, R_e, R_f$, y el siguiente mensaje *ICMP* con $TTL = 7$ sigue el camino $H_1, R_a, R_n, R_b, R_c, R_d, R_e, R_f$; Se obtiene a la salida del *traceroute* que el *router* R_f aparezca dos veces con el mismo valor de *q-ttl*, pero la posición n incrementó en 1 respecto a la posición anterior, aunque sea el mismo *router*. En el ejemplo se tendría *q-ttl* $= 3$ para $n = 3$ y *q-ttl* = 3 para $n = 4$ para el mismo *LSR* R_f. Es decir, la variación *q-ttl* $= n - \Delta$ se produce porque el algoritmo que genera la posición n, no tiene capacidad de detectar que un mensaje *ICMP* puede expirar en el mismo *router* para distintos valores de *TTL* debido al balanceo de carga. Sin embargo este desfase es escaso (ver figura 3.4) y no genera errores considerables en la estimación de la posición n. En este caso el balanceo de carga se produce al exterior del túnel *MPLS*.

Se observó también que en otros casos, dos *LSRs*, esta vez distintos, pueden aparecer en el *traceroute* con un mismo valor de *q-ttl*. En este caso se cree que se debe a una política de balanceo de carga al interior del túnel *MPLS*. Este comportamiento ocasionado por balanceadores de carga *per-flow* se observó en menos del $0,5\,\%$ de los túneles MPLS encontrados.

q-ttl = 1: Se puede obtener un *traceroute* con una salida del tipo R_a, R_b, R_c, $R_{d/mpls}^{q\text{-}ttl=1}$, $R_{e/mpls}^{q\text{-}ttl=1}$, $R_{f/mpls}^{q\text{-}ttl=1}$, $R_{g/mpls}^{q\text{-}ttl=4}$. El *LSR* $R_{f/mpls}^{q\text{-}ttl=1}$ ocupa la posición $n = 3$ y sin embargo le corresponde un *q-ttl* = 1. El valor del *q-ttl* en este caso no representa la posición del *LSR* dentro del túnel y significa que el TTL_{ip} o *q-ttl* ha sido actualizado al valor del TTL_{lse} antes de formar el mensaje *ICMP* `time-exceeded`, estableciéndose en 1. Es decir, el número de firmas con *q-ttl* $= 1$ cuando $n > 1$

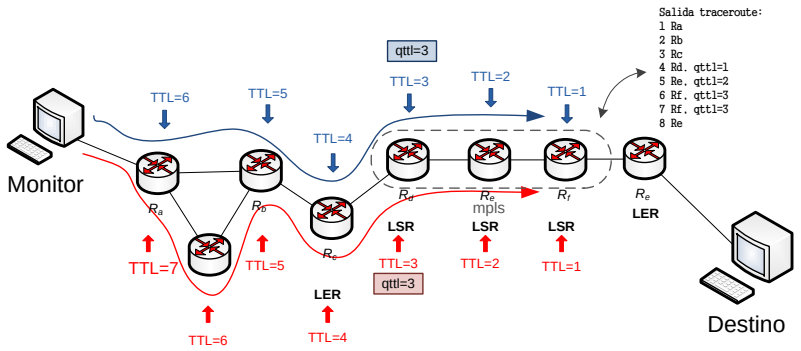


FIGURA 3.5: En azul se muestra el recorrido de un mensaje de *traceroute* originada con $TTL = 6$, y en rojo un mensaje con $TTL = 7$. El balanceo de carga ocasiona que se produzca un desfase entre la posición n y el *q-ttl*.

representa la cantidad de *LSRs* donde la firma *q-ttl* no es válida. El hecho de que esta firma no se considere válida, no significa que no está activa la opción `ttl-propagate`, significa que en algunos fabricantes el *ICMP* `time-exceeded` se forma actualizando el TTL_{ip} del paquete que acaba de expirar dentro del túnel. Se encontró que aproximadamente en el 85 % de los túneles explícitos se puede identificar también la firma *q-ttl*, mientras que en el 15 % restante la firma *q-ttl* siempre vale 1, sin aportar información sobre la posición del túnel.

El análisis entre los valores de *q-ttl* y n permite dos conclusiones:

i la posición de n varía respecto al valor del *q-ttl* principalmente por la incapacidad de detectar el primer *LSR*.

ii y el valor de *q-ttl* puede ser 1 aún cuando el *LSR* no sea el primero del túnel *MPLS*, lo que impide tener un valor real de la ubicación del *LSR* en el túnel usando el valor de la firma.

Finalmente, como se observa en la figura 3.4, los casos en donde $q\text{-}ttl = 1$ y donde la firma *q-ttl* no aporta información de la posición dentro del túnel, son escasos en comparación

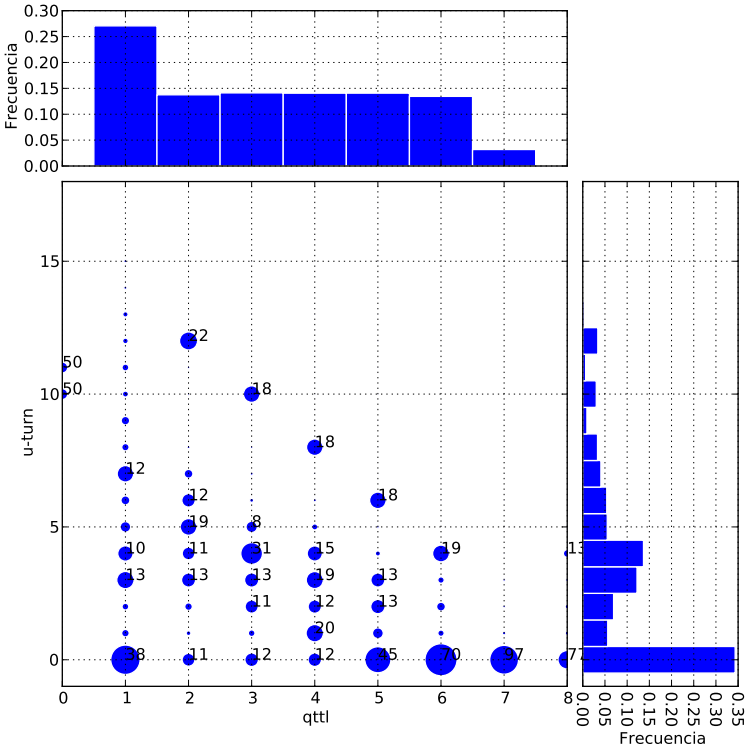
a los casos donde se obtuvo un n desfasado en +1, obteniendo una posición errada. En adelante se considerará que el valor del *q-ttl* es el más confiable para revelar la posición de un *LSR*, si no se dispone de esta firma se puede usar el valor n, sabiendo que en el peor de los casos su valor estará atrasado o adelantado respecto en una posición respecto al valor real. Esta conclusión podría resultar obvia, sin embargo el presente análisis permitió conocer el comportamiento de la posición n, y conocido su comportamiento, se lo utilizará posteriormente para evaluar los resultados de las firmas en aquellos *LSRs* dónde el valor *q-ttl* no puede obtenerse (ver sección 3.3.4.3).

3.3.4.3. Comportamiento de la firma *u-turn* en el túnel *MPLS*

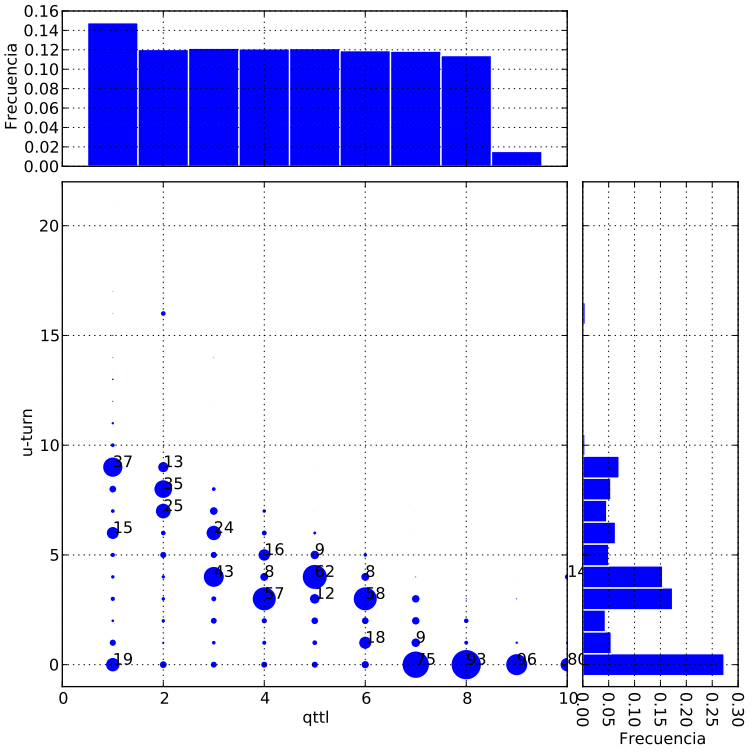
En esta sección se analiza la validez de la firma *u-turn* comparándola con la firma *q-ttl* obtenida de los *LSRs* tanto explícitos como implícitos. También se la compara con la posición n que el *LSR* ocupa en el túnel *MPLS* en los casos donde el *q-ttl* no pudo ser obtenido. El primer análisis, *u-turn* en función de *q-ttl* tiene por objeto obtener una idea del comportamiento general de la firma *u-turn* en aquellos *LSR* que presentaron la firma *q-ttl*. El segundo análisis, *u-turn* en función de n tiene por objeto analizar el comportamiento de la firma *u-turn* en aquellos *LSRs* dónde ninguna otra firma pudo ser obtenida, siendo el único valor de referencia la posición n en la que el *LSR* se descubre al interior del túnel.

La figura 3.6 muestra los resultados del análisis propuesto para túneles con longitud seis y ocho. Idealmente los valores de *u-turn* deberían estar relacionados mediante la expresión $u\text{-}turn = -2q\text{-}ttl + L$ donde L representa la longitud del túnel. Sin embargo se encontró que los valores de *u-turn* se encuentran dispersos, llegando a marcar solamente una ligera tendencia. Se cree que este comportamiento observado se debe a que no existe ninguna garantía para asegurar que los mensajes `time-exceeded` y `echo-reply` utilizados para obtener la firma *u-turn* sigan un mismo camino de vuelta. Con lo cual el valor de *u-turn* no representa directamente la existencia de un túnel *MPLS* sino también la existencia de balanceadores en el camino de regreso. Para los túneles con longitud menor a seis, se encontró que la dispersión entre *u-turn* y *q-ttl* es aún mayor.

La figura 3.7 muestra los resultados del análisis *u-turn* en función de la posición n. Se observa que los valores son aún más dispersos y no existe ninguna relación clara entre las dos variables. Esta observación permite concluir que no existe una prueba consistente de



(a) *u-turn* vs *q-ttl*, l=6



(b) *u-turn* vs *q-ttl*, l=8

FIGURA 3.6: Dispersión entre los valores *u-turn* vs *q-ttl* para distintas longitudes del túnel *MPLS*. Se observa que los valores presentan menor dispersión cuando aumenta la longitud del túnel, sin embargo en ningún caso los valores tienden hacia el comportamiento ideal $u\text{-}turn = -2q\text{-}ttl + L$.

que la firma *u-turn* infiera correctamente un *LSR* dentro de un túnel *MPLS*, pues su valor bien puede deberse a la existencia de un túnel o bien a que los mensajes `time-exceeded` y `echo-reply` siguieron caminos distintos debido al balanceo de carga.

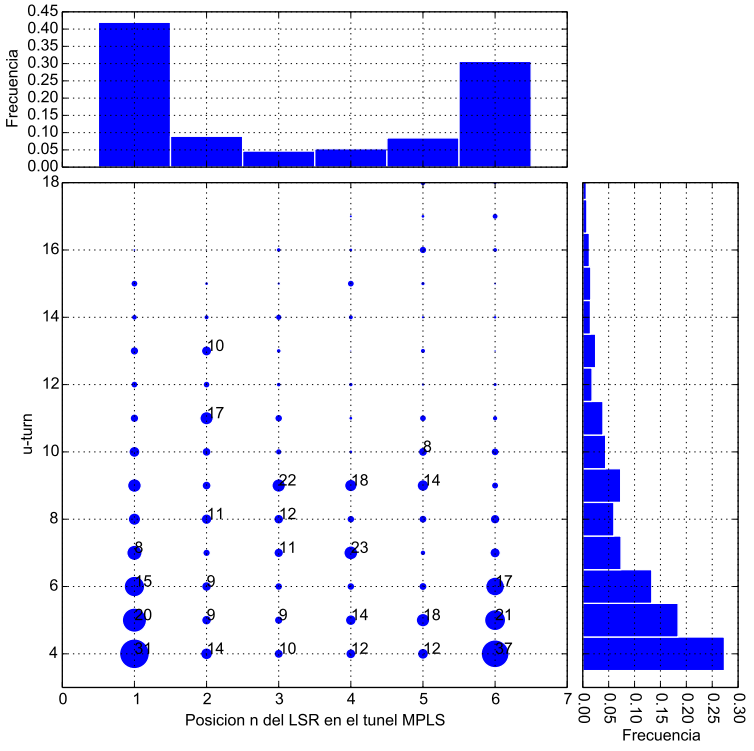
Los análisis hasta ahora planteados permiten concluir que la firma *u-turn* es demasiado imprecisa, y en vista de los escasos túneles encontrados mediante este método (se la encontró en cerca del $2,8\,\%$ de *paris-traceroutes* analizados), se recomienda no usarla en trabajos posteriores. La firma ya advierte su imprecisión cuando se la compara entre los valores de *q-ttl* y los valores de *u-turn* en aquellos *LSR* que permitieron obtener estas dos firmas. Y finalmente la firma carece totalmente de sentido cuando se la compara con la posición n en aquellos *LSR* que solamente se identificaron usando esta firma, y que es donde realmente la firma resulta útil en caso de que sea válida.

Esta imprecisión encontrada en la firma *u-turn* hizo que se considere un valor de *u-turn* $>$ 4 en lugar de *u-turn* > 3 como se recomienda en [DLMP12], con el objetivo de realizar un filtro más selectivo esperando disminuir los errores encontrados, con lo cual la firma *u-turn* se redujo de encontrarse en el $5,7\,\%$ de *paris-traceroutes*, a encontrarse en el $2,8\,\%$. Se decidió seguir usando la firma *u-turn*, en lugar de descartarla, esperando que en los análisis posteriores se presente algún indicio de los errores acarreados por esta firma.

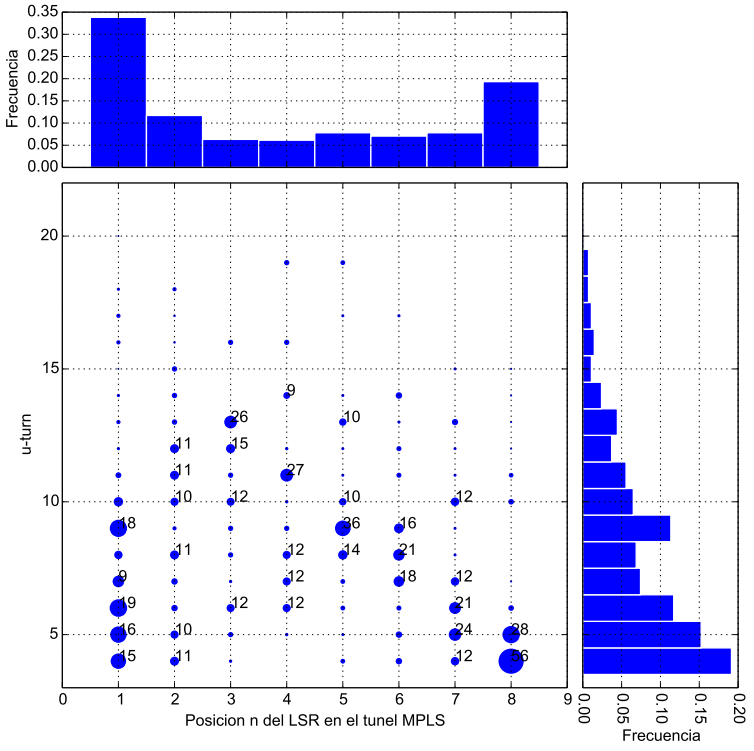
3.4. Formación de Grafos

Con el objetivo de analizar el impacto de los túneles *MPLS*, se comparó las topologías obtenidas antes y después de que los túneles sean identificados. La presente sección tiene por objeto describir el proceso involucrado en la obtención de los distintos grafos a partir de la herramienta de exploración. Primero se describe el proceso de construcción del grafo *IPv4*, posteriormente se describe el proceso de Resolución de Alias para obtener un grafo de *routers* y finalmente se describe la formación del grafo diferenciando en él los enlaces *MPLS*.

La tabla 3.3 resume el tamaño de los distintos grafos obtenidos.



(a) *u-turn* vs n, l=6



(b) *u-turn* vs n, l=8

FIGURA 3.7: Dispersión entre la posición n y el valor *u-turn* para distintas longitudes del túnel *MPLS*. No hay relación entre la posición del *LSR* dentro del túnel y el valor de la firma *u-turn*.

Grafo	**Descripción**	**Vértices**	**Aristas**	**<Grado>**
G_{ip}	Grafo IP	90882	508867	$11,12$
G_r	Grafo de *routers*	76142	338685	$8,90$
$G_{r\setminus mpls}$	Grafo de *routers* con contracción *MPLS*	67967	297321	$8,75$
G_{mpls}	Grafo general *MPLS*	9297	12523	$2,70$
$G_{mpls}^{inter\text{-}as}$	Grafo de enlaces *MPLS* inter-AS	234	352	$3,01$

TABLA 3.3: Tamaño de los distintos grafos analizados.

3.4.1. Grafo IPv4 G_{ip}

El Grafo *IPv4* se define como $G_{ip} = (V_{ip}, E_{ip})$, y se obtiene a partir de las salidas del *paris-traceroute*. El único procesamiento que reciben los datos en este punto es la resolución de interfaces que no responden.

Las interfaces que no responden se resuelven siguiendo el proceso expuesto en la sección 2.2.2 pero con una modificación: basta encontrar 2 *paris-traceroutes* $t_{A->B}$ y $t_{C->D}$ que contengan la *IP* predecesora y la misma *IP* sucesora a la interfaz que sin respuesta "*", para que se le asigne un mismo nombre *ur.1*. Se decidió ignorar la condición de "destino común" (ver sección 2.2.2) porque se observó que generaban tantos *routers ur.n* como *traceroutes* con destino distinto existían, de modo que no se reducía de forma considerable la cantidad de interfaces sin respuesta aún cuando se tenía una misma *IP* predecesora y una misma *IP* sucesora.

Una vez resueltas las *IPs* sin respuesta, el grafo *IPv4* G_{ip} se forma siguiendo las siguientes premisas:

i Dado un *traceroute* t de n saltos definido como $t = IP_1, IP_2, .., IP_{n-1}, IP_n$, cada una de las *IP* forman los vértices del grafo, de modo que $IP_1, IP_2, ..., IP_{n-1}, IP_n \in V_{ip}$ y las aristas del grafo se forman entre aquellas *IP* consecutivas de modo que $(IP_1, IP_2), ..., (IP_{n-2}, IP_{n-1}), (IP_{n-1}, IP_n) \in E_{ip}$.

ii Si existe una interfaz r^* que no responde al *traceroute*, de modo que $r^* \in t$ y a su vez no ha podido resolverse en el proceso de resolución de interfaces que no responden, entonces se ignora cualquier adyacencia con r^*.

Si bien al ignorar las adyacencias con las interfaces que no responden se pierde información, es información no crítica. Las interfaces ignoradas no pueden ser usadas en ningun

procesamiento de información como la Resolución de Alias o la detección de túneles *MPLS* y formarán nodos solamente de grado 2, con lo cual no afectan considerablemente a los objetivos de este trabajo.

La Tabla 3.3 resume el tamaño del Grafo *IP* G_{ip} obtenido en este proceso.

3.4.2. Grafo de *routers* G_r

El Grafo de *routers* G_r está formado por vértices que representan *routers*, los cuales contienen una o más direcciones *IP*. Se obtiene a partir de la Resolución de Alias y de resolver las interfaces que no responden. Estos procesos se explican de forma detallada en la sección 2.2.3 y 2.2.2 respectivamente. En la presente sección se explicarán las consideraciones particulares que se aplicaron a cada uno de los procesos.

3.4.2.1. Resolución de Alias

El proceso de Resolución de Alias básicamente consiste en resolver cuales de las direcciones *IP* descubiertas mediante *paris-traceroute* pertenecen al mismo *router*. Este proceso involucra una serie de pasos y condiciones previamente detalladas en la sección 2.2.3 y que se resumen a continuación:

i **Creación de subredes candidatas**. Se crearon todas las subredes posibles a partir de la máscara 27, es decir: $S_m \geq 27$, donde S representa la subred y m su máscara [GS09].

ii **Selección de subredes**. De todas las subredes previamente creadas se eligió aquellas que cumplen con la condición de Precisión y $C \geq 0,5$, donde C representa el factor de completitud [GS09].

iii **Orden de procesamiento**. Se procesan empezando por las subredes con mayor factor de completitud. Si este factor coincide en varias subredes, se da prioridad a la subred cuyas direcciones *IP* hayan aparecido en la mayor cantidad de *paris-traceroutes* [GS09].

iv **Condiciones para inferir un alias**. Un alias se considerará verdadero solamente si cumple con la condición "Sin Bucle" y con la condición "Vecinos en Común" sin importar que sea un enlace multi-acceso o punto a punto [GS09].

Con respecto al punto *iv*, la bibliografía recomienda aplicar la condición "Vecinos en Común" solamente en los enlaces multi-acceso e ignorar esta condición cuando los candidatos a alias se forman a partir de redes punto a punto o subredes con máscaras mayores a 30. Sin embargo se encontró que esto genera una elevada cantidad de errores, de forma que se consideraban alias, *IPs* que incluso se encontraban en regiones geográficas distintas, esto se verificó usando herramientas basadas en *whois* y *nsloopback*, observando que *IPs* tratadas como un alias pertenecían a *routers* resueltos con nombres distintos. Para evitar los alias inferidos erróneamente se decidió que la condición "Vecinos en Común" se cumpla también para los enlaces punto a punto. Se cree que este error en los alias se debe a un error acarreado desde el momento de la creación de las subredes, pues el hecho de que una subred adquiera la máscara adecuada depende básicamente del factor de completitud. Este factor indica la cantidad de *IPs* asignadas dentro de la subred y se elige aquel con completitud mayor, por tanto depende también de la exploración, pudiéndose dar el caso de que se encuentren tan pocas *IPs* de una subred de máscara m(por limitaciones de la exploración), que el factor de completitud favorecerá a la subred con máscara n tal que $n > m$. De este modo, se selecciona una subred erróneamente con máscara mayor a la real.

La tabla 3.3 muestra un resumen del tamaño del Grafo de *routers* G_r.

3.4.3. Grafo *MPLS* G_{mpls}

El grafo **MPLS** G_{mpls} se define como un grafo formado por las aristas *MPLS* E_{mpls} y los vértices *MPLS* V_{mpls}.

i **Vértices *MPLS*** V_{mpls}: Se define así al conjunto de *routers* en donde se ha encontrado al menos una dirección *IP* asociada a un *LSR*, cualesquiera sea el tipo de túnel que dicho *LSR* forma: implícito, explícito u opaco. En vista de que un *router* puede contener varias direcciones *IP*, algunas asociadas a un túnel *MPLS* y otras no, un vértice *MPLS* puede formar adyacencias tanto con otros vértices *MPLS* como con vértices *IP*.

ii **Aristas *MPLS*** E_{mpls}: Son las aristas formadas por cualquier par de vértices $(V^a_{mpls}, V^b_{mpls}) \in E_{mpls}$ de forma que $V^a_{mpls} y V^b_{mpls} \in V_{mpls}$.

Es decir, el grafo G_{mpls} no es más que un subgrafo de G_r, inducido solamente por los vértices de G_r que representan a un *LSR* o *router MPLS*.

La tabla 3.3 muestra un resumen del tamaño del Grafo de *routers* G_{mpls}.

3.4.4. Grafos inducidos por vértices pertenecientes a un mismo Sistema Autónomo

i G_{ip}^{as}: Es el grafo inducido de G_{ip}, tal que $G_{ip}^{as} = (V_{ip}^{as}, E_{ip}^{as})$, donde todos los vértices $v_{ip}^{as} \in V_{ip}^{as}$ pertenecen a un mismo sistema autónomo as.

ii G_r^{as}: Es el grafo inducido de G_r, tal que $G_r^{as} = (V_r^{as}, E_r^{as})$, donde todos los vértices $v_r^{as} \in V_r^{as}$ pertenecen a un mismo sistema autónomo as.

iii G_{mpls}^{as}: Es el grafo inducido de G_{mpls}, tal que $G_{mpls}^{as} = (V_{mpls}^{as}, E_{mpls}^{as})$, donde todos los vértices $v_{mpls}^{as} \in V_{mpls}^{as}$ pertenecen a un mismo sistema autónomo as. En el presente trabajo, cada una de las componentes conexas del grafo G_{mpls}^{as} recibirá el nombre de Dominio *MPLS*.

3.4.5. Grafo de *routers* con contracción de nodos *MPLS* $G_{r\setminus mpls}$

El Grafo $G_{r\setminus mpls}$ representa al grafo G_r modificado. Esta modificación consiste en identificar en G_r todos los Dominios *MPLS* de los distintos sistemas autónomos y contraerlos a un solo vértice V_c. De esta forma se crearan tantos vértices V_c como componentes conexas de túneles *MPLS* existan perteneciendo a un mismo sistema autónomo. Antes de continuar con los detalles involucrados en el proceso de contracción de nodos *MPLS*, en las siguientes secciones se definen algunos conceptos que serán posteriormente utilizados.

3.4.5.1. Contracción de nodos *MPLS*

En la presente sección se describe el proceso utilizado para la creación del grafo $G_{r\setminus mpls}$. La contracción en teoría de grafos se define como la operación que elimina una arista del grafo al mismo tiempo que fusiona los dos vértices extremos. La contracción de nodos *MPLS* en este caso tiene consiste en eliminar del grafo $G_r = (V_r, E_r)$ todas las aristas $e_{mpls}^{as} = (v_{mpls}^{a}, v_{mpls}^{b}) \in E_{mpls} \subseteq E_r$ tal que $v_{mpls}^{a}, v_{mpls}^{b} \in V_{mpls} \subseteq V_r$ pertenezcan al

mismo sistema autónomo as, y fusionar los vértices v^a_{mpls}, v^b_{mpls} en un nuevo vértice $v^{contraido}_{mpls} \in V_r$. Este proceso se realiza continuamente hasta que no quedan más aristas $e_{mpls} \in E_{mpls} \subseteq E_r$, obteniendo como resultado que el grafo $G_{r\setminus mpls}$.

En este proceso se crean tantos vértices contraídos como Dominios *MPLS* se encuentren en la topología. La contracción de los vértices se realiza solamente entre aquellas aristas cuyos vértices forman parte de un mismo sistema autónomo y permite analizar si las propiedades propias de la topología *IP* se modifican con la precencia de túneles *MPLS*.

La tabla 3.3 muestra el tamaño del grafo $G_{r\setminus mpls}$ obtenido de este proceso.

Se define también el grafo $G^{as}_{r\setminus mpls}$, que no es más que el grafo inducido de $G_{r\setminus mpls}$, tal que $G^{as}_{r\setminus mpls} = (V^{as}_{r\setminus mpls}, E^{as}_{r\setminus mpls})$, donde todos los vértices $v^{as}_{r\setminus mpls} \in V^{as}_{r\setminus mpls}$ pertenecen a un mismo sistema autónomo.

3.4.6. Grafo de *ASs* conectados por túneles *MPLS* $G^{inter\text{-}as}_{mpls}$

Es el grafo $G^{inter\text{-}as}_{mpls} = (V_{as}, E^{inter\text{-}as}_{mpls})$, donde cada vértice $v \in V_{as}$ representa un sistema autónomo y cada una de las aristas $(e^{as_i}_{mpls}, e^{as_j}_{mpls}) \in E^{inter\text{-}as}_{mpls}$ representa a un túnel o enlace *MPLS* existente entre dos *ASs*.

La tabla 3.3 muestra el tamaño del grafo $G^{inter\text{-}as}_{mpls}$ obtenido de este proceso.

3.5. Análisis de los Grafos obtenidos

En esta sección se comparan las distintas métricas de la topología obtenida a través de las diferentes etapas de este trabajo. Se identifican así cuatro grafos:

i **Grafo *IPv4*** G_{ip}, definido en la sección 3.4.1.

ii **Grafo de *routers*** G_r, definido en la sección 3.4.2.

iii **Grafo de *routers* con contracción de nodos *MPLS*** $G_{r\setminus mpls}$, definido en la sección 3.7.1.2.

iv **Grafo *MPLS*** G_{mpls}, definido en la sección 3.4.3.

El grafo $G_{mpls}^{inter\text{-}mpls}$ se analizará posteriormente en la sección 3.7.2.

En la presente sección se explica también el concepto de *binnings*, proceso que fue usado sobre cada una de las métricas de los grafos analizados.

3.5.1. *Binning* logarítmico

Binning es el proceso de promediar los valores que caen en determinado contenedor (del inglés *bin*), por ejemplo: valores que caen en un rango de k. Los valores promediados eliminan las oscilaciones de las mediciones que pueden interpretarse como ruido. Cuando el tamaño del contenedor k no es fijo y se incrementa logarítmicamente, entonces el proceso recibe el nombre de *binning* logarítmico. El número de bins B o número de contenedores dependerá del número de muestras que se deseen promediar: si B es muy pequeño, es decir si existen pocos contenedores y por tanto el intervalo k es grande, el proceso de *binning* puede resultar dañino ya que los valores se promediarán en intervalos demasiado grandes, perdiendo demasiada información; por otro lado, si B es muy grande, esto es que existen demasiados contenedores y el valor del intervalo k es muy chico, entonces el proceso de *binning* resulta inútil, pues la información continúa mostrándose con ruido. El *binning* logarítmico es útil para distribuciones leyes de potencia, en donde la cantidad de muestras disminuye para valores elevados, resultando más eficiente aumentar el tamaño del contenedor k para los rangos de valores con pocas muestras y disminuir el tamaño del contenedor para rangos de valores con elevada cantidad de muestras.

En el presente trabajo se usaron *binnings* logarítmicos para una mejorar visualización de los resultados. Para la linealización de las curvas obtenidas, se usó la herramienta plfit[3], software basado en el algoritmo descrito en [CSN09].

3.5.2. Distribución de Grados

La figura 3.8 muestra que los grafos G_{ip}, G_r y $G_{r\setminus mpls}$ presentan aproximadamente la misma distribución de grado, sin que este métrica se vea afectada por los procesos de Resolución de Alias y Contracción *MPLS*. Aproximando la distribución de grado a la ecuación $P(g) \approx g^{-\gamma}$, se obtuvo que $\gamma \approx 3,24$.

[3] https://github.com/ntamas/plfit

Por otro lado se observa que la distribución de grados del grafo G_{mpls} decae con mayor rapidez y alcanza un grado máximo de 36. En este caso se obtuvo $\gamma \approx 4{,}26$.

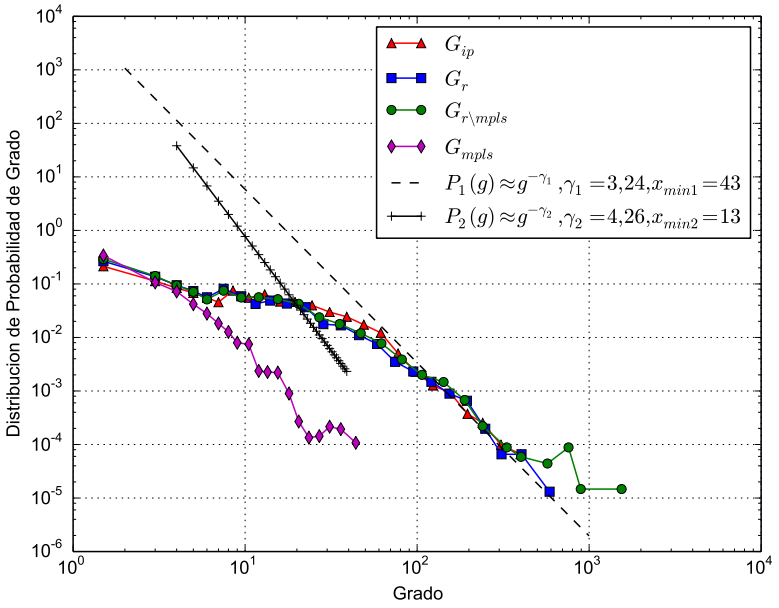


FIGURA 3.8: La figura muestra la Función Distribución de Probabilidad de Grado. Para disminuir el ruido presente en la cola de la curva, se usó bins logarítmicos $B = 30$.

3.5.3. Distribución del Grado Medio de los Vecinos

La figura 3.9 muestra la Distribución del Grado Medio de los Vecinos. En este caso el proceso de contracción *MPLS* afecta levemente el grado medio de los vecinos del grafo $G_{r\setminus mpls}$. Los grafos G_{ip} y G_r presentan un ligero comportamiento concordante. Se observa también que el proceso de contracción *MPLS* afecta ligeramente a los nodos de grado menor a 40, este comportamiento es esperado, puesto que como se muestra en la figura 3.8, el grado máximo del grafo G_{mpls} se encuentra alrededor de este orden. Finalmente el grafo G_{mpls} presenta un comportamiento plano para cualquiera de sus grados.

3.5.4. Distribución del Coeficiente de *Clustering*

La figura 3.10 muestra la Distribución del Coeficiente de *Clustering*. Se observa que este parámetro se modifica levemente por el proceso de Resolución de Alias y por el proceso

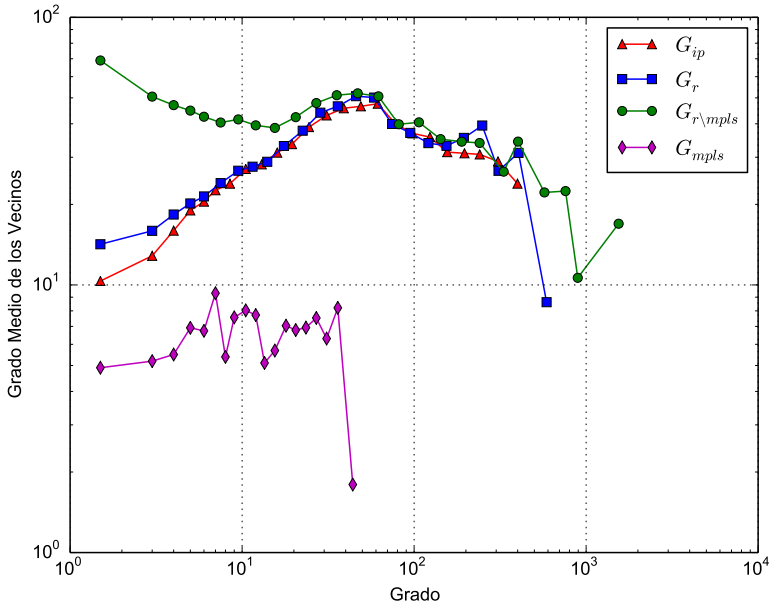


FIGURA 3.9: Distribución del grado medio de los vecinos.

de contracción *MPLS*. Este comportamiento se explica en la contracción de vértices involucrado en ambos procesos. En el caso de Resolución de alias, varias *IPs* se contraen en un mismo nodo que representa un *router*. En el caso de la contracción *MPLS*, todas las componentes conexas *MPLS* se contraen en un solo vértice. Ambos procesos colaboran con ello a que se reduzca la cantidad de vértices en el grafo pero además a aumentar la densidad de aristas entre los vértices, pues se eliminan vértices pero no aristas. En la figura también se observa que a diferencia de la Distribución del Grado Medio de los Vecinos, el Coeficiente de Clustering afecta por igual a todos los nodos del grafo $G_{r\backslash mpls}$ independientemente de su grado, esto significa que existen redes *MPLS* contra las que se conectan por igual tanto los vecinos de nodos de grado bajo como los de grado alto. La pendiente negativa significa que los nodos de bajo grado tienen más vecinos altamente conectados entre sí que los nodos de alto grado, comportamiento ligeramente discordante. Otra observación importante resulta al destacar que el Coeficiente de *Clustering* no se ve afectado significativamente por el proceso de contracción *MPLS*. Este comportamiento tiene sentido, puesto que los vértices del grafo G_r se vieron afectados por el aumento del grado promedio que causó la contracción de nodos, pero el número de aristas del grafo G_r no aumenta y por tanto el coeficiente de *Clustering* tampoco.

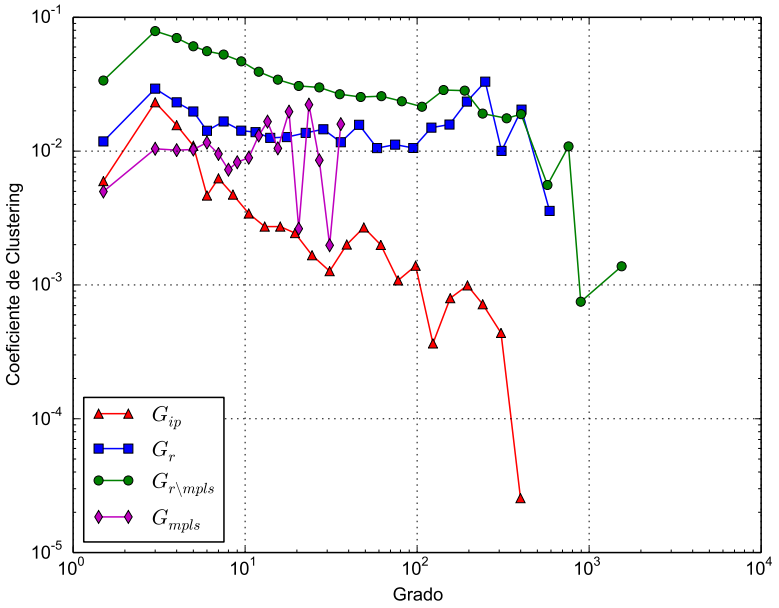


FIGURA 3.10: Coeficiente de Clustering.

3.6. Visualización de los Grafos obtenidos

En las figuras 3.11, 3.12 y 3.13 se muestra la visualización usando el método de *k-núcleos* para los grafos G_{ip}, G_r y $G_{r\setminus mpls}$ respectivamente. Se observa que para todos los grafos analizados se mantiene estable el número de capa y la forma de distribución de los nodos alrededor del *núcleo central*. Sin embargo, se observa que el algoritmo de k-núcleos genera varias componentes desconexas. Se sospecha que estas componentes surgen debido a que el presente trabajo no considera los enlaces *IPv6*.

Por otro lado, la visualización del grafo $G_{r\setminus mpls}$ (ver 3.13) muestra que los nodos de mayor grado se encuentran distribuidos en las capas ubicadas entre el núcleo central y la periferia, a diferencia de los grafos G_{ip} y G_r, donde la mayoría de nodos con grados elevados se ubican solamente en el núcleo central o sus cercanías. Estos nodos de grado elevado representarían los dominios *MPLS* que se contrajeron a un solo vértice. Se observa también que el proceso de Contracción *MPLS* disminuye la cantidad de vértices que se ubicaban en las capas intermedias respecto del grafo G_r, esto significaría que los *LSRs* se ubican generalmente entre estas capas intermedias, mientras que las capas periféricas y centrales representan en general a vértices *IP*. Otra observación que apoya esta conclusión, es la aparición de nodos de alto grado en las capas intermedias después

del proceso de contracción *MPLS*. El hecho de que los Dominios *MPLS* aparezcan en las capas intermedias (ver los círculos rojos en la figura 3.13) con un grado elevado puede significar que estos nodos actúan de concentradores de nodos de menor grado (capas periféricas), y que los nodos *IP* de alto grado (capas centrales), están más conectados entre sí con otros nodos *IP* también de alto grado, y no con nodos *MPLS*, siendo este el motivo por el cual las capas centrales no presentan cambios considerables antes y después del proceso de contracción *MPLS*.

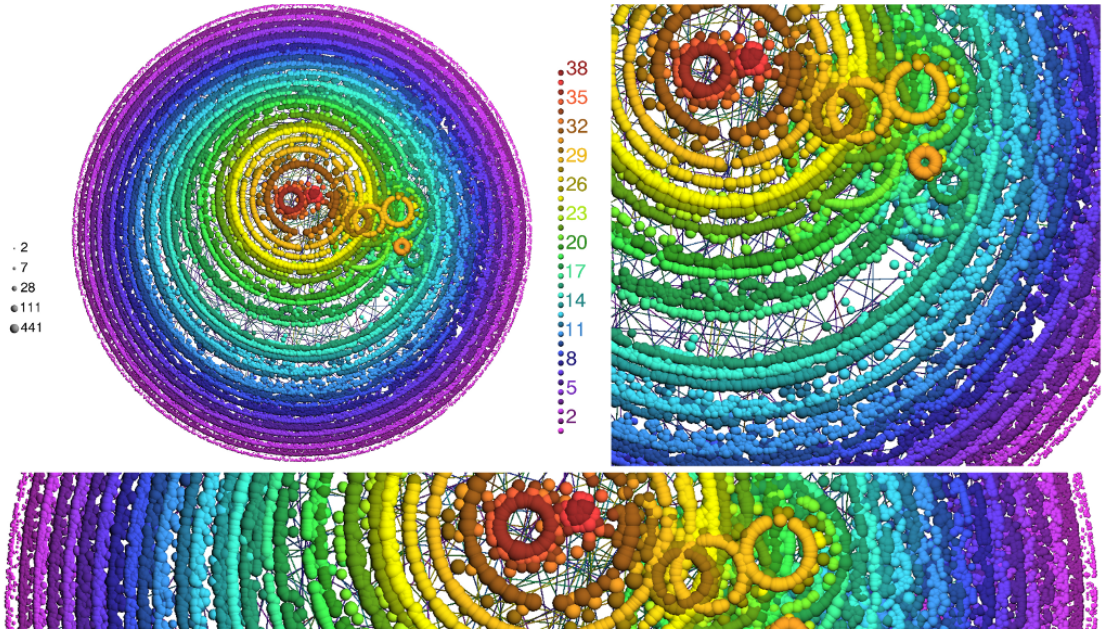


FIGURA 3.11: Visualización usando la descomposición de *k-núcleos* del grafo *IP* G_{ip}.

3.7. Enlaces *MPLS* y Sistemas Autónomos (ASs)

En esta sección se analiza el comportamiento de los enlaces *MPLS* dentro de un mismo sistema autónomo y hacia otros sistemas autónomos. Se usarán los términos intra-AS e inter-AS para definir si un enlace *MPLS* conecta *LSRs* pertenecientes al mismo o diferentes sistemas autónomos:

i Se define como un enlace *MPLS* intra-AS a todo enlace existente entre dos *LSRs* que registren el mismo número de sistema autónomo.

ii Se define como un enlace *MPLS* inter-AS a todo enlace existente entre dos *LSRs* que registren diferente número de sistema autónomo.

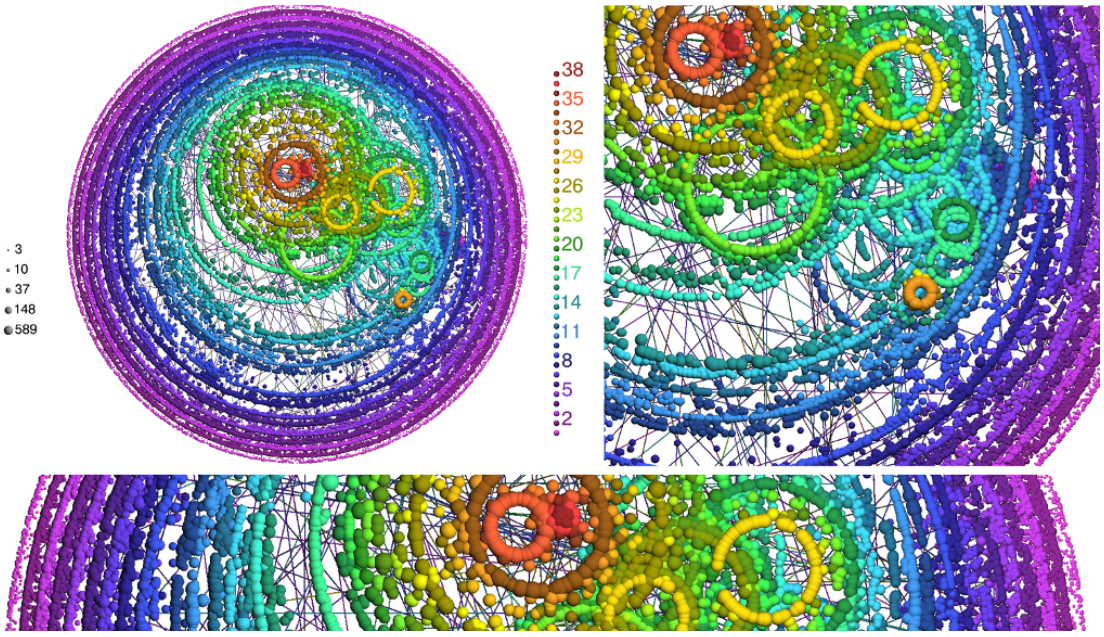


FIGURA 3.12: Visualización usando la descomposición de *k-núcleos* del grafo de *routers* G_r.

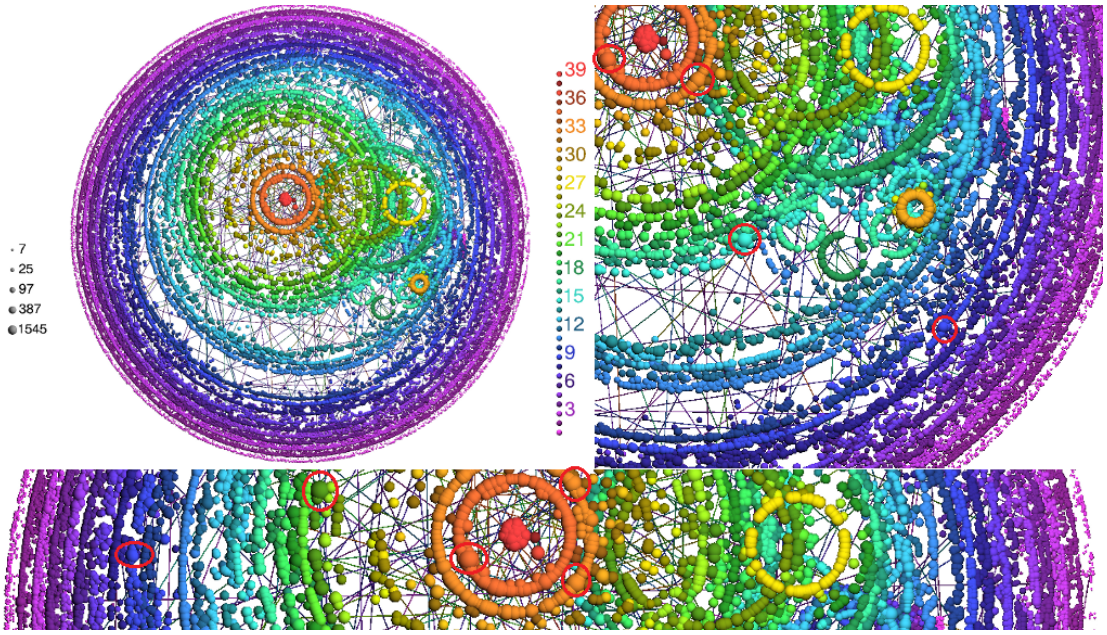


FIGURA 3.13: Visualización usando la descomposición de *k-núcleos* del grafo *routers* con contracción *MPLS* $G_{r\setminus mpls}$. Se marcaron en círculos rojos algunos nodos cuyo grado es mayor al del resto de nodos pertenecientes a la misma capa, estos nodos indican la presencia de regiones *MPLS*.

Se encontró que los túneles *MPLS* ocurren con mayor frecuencia entre *LSRs* pertenecientes a un mismo sistema autónomo. Solamente el $13,65\,\%$ de las vecindades entre *LSRs* fueron inter-AS, es decir entre *LSRs* pertenecientes a *ASs* distintos.

3.7.1. Enlaces *MPLS* intra-*AS*

En esta sección se analiza los enlaces *MPLS* intra-AS, es decir aquellos cuyos nodos registran un mismo número de sistema autónomo. El objetivo de esta sección es entender el comportamiento de los túneles *MPLS* dentro de un mismo *AS*.

3.7.1.1. Comparación entre G_r^{as} y G_{mpls}^{as}

En la sección 3.4.4 se definió el grafo inducido *MPLS* G_{mpls}^{as}, que no es más que un grafo formado por los túneles *MPLS* que se ubican dentro de un mismo *AS* (enlaces intra-*AS*), de aquí se obtuvieron tantos grafos *MPLS* G_{mpls}^{as} como sistemas autónomos con túneles *MPLS* se encontraron en su interior. Ya se analizó previamente cómo estos dominios *MPLS* impactan la topología de *routers* (ver sección 3.7.1.2). En la presente sección se describe el comportamiento de los túneles *MPLS* dentro de su propio sistema autónomo, ya que es donde se los observaron con mayor frecuencia, y se lo compara con el grafo G_r^{as}. Se encontró que el $86,35\,\%$ de los enlaces *MPLS* ocurren dentro de un mismo sistema autónomo.

La figura 3.14 representa un histograma de la fracción de cantidad de enlaces *MPLS* sobre la cantidad total de enlaces descubiertos por sistema autónomo ($r = links_{mpls}/links_{totales}$). La figura permite conocer que no hay una tendencia clara en el despliegue de enlaces *MPLS* por sistema autónomo, es decir cada *AS* es muy particular en cuanto a la fracción de enlaces *MPLS* que se descubrieron en su interior.

Continuando con el análisis de los túneles *MPLS* dentro de un *AS* se proponen las figuras 3.15 y 3.16 (los detalles de los grafos de estas figuras se muestran en la tabla 3.5), que comparan los resultados de la visualización basada en *k-núcleos* para algunos de los *ASs* con más enlaces *MPLS*. Se observa que hay una tendencia a que el núcleo central de la red *MPLS* en cada *AS*, se forme por varios nodos, inclusive llegando a tener una cantidad de nodos en el núcleo central comparable con la de las periferias. Esta observación permite concluir que los *routers MPLS* están más conectados entre sí que los *routers IP*. Se

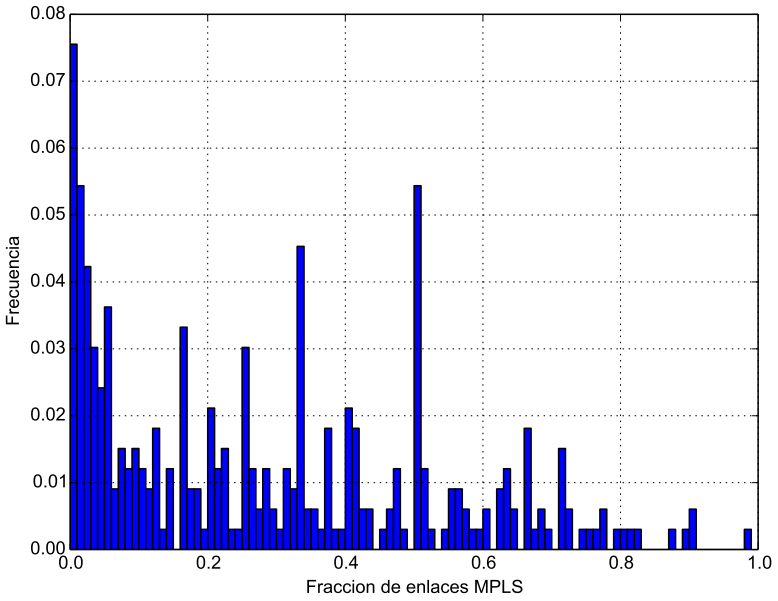


FIGURA 3.14: Histograma de la fracción de enlaces *MPLS* sobre el total de enlaces descubiertos en un *AS*. El eje horizontal representa la fracción $r = links_{mpls}/links_{totales}$. El eje vertical representa la frecuencia con la cual se observó sistemas autónomos con valores cercanos a r.

observa también que en el AS33363 (Figura 3.16) la descomposición en *k-núcleos* genera componentes desconexas, lo que sería un indicador de la existencia de varios *backbones MPLS* dentro de un mismo *AS*.

3.7.1.2. Comparación entre $G^{as}_{r\setminus mpls}$ y G^{as}_{r}

En la sección 3.7.1.2 se analizó la topología de *Internet* mediante la contracción de un dominio *MPLS* a un solo vértice siempre y cuando pertenezcan a un mismo *AS*, esto permitió obtener un panorama general de como se distribuyen las redes *MPLS* sobre el grafo de Internet. En la presente sección se repite la misma experiencia pero en lugar de analizar el grafo total de Internet, se analiza lo que pasa dentro de un mismo sistema autónomo cuando se realiza la contracción *MPLS*, mediante el grafo $G^{as}_{r\setminus mpls}$. Para esto se analizaron cuatro sistemas autónomos, los cuales corresponden los *AS* con más enlaces *MPLS* en su interior, previamente listados en la tabla 3.4.

La figura 3.17 muestra la visualización basada en k-núcleos de los grafos $G^{as}_{r\setminus mpls}$, donde los *AS* se corresponden a los 4 con mayor enlaces *MPLS*. Los detalles del tamaño de

	Número de AS	**Nombre de AS**	**Enlaces *MPLS* descubiertos**	**Fracción**
1	3356	LEVEL3 Level 3 Communications	3692	0, 30
2	7018	ATT-INTERNET4 - AT&T Services, Inc.	3596	0, 59
3	6830	LGI-UPC Liberty Global Operations B.V.	1530	0, 63
4	33363	BHN-TAMPA - BRIGHT HOUSE NETWORKS, LLC	955	0, 86
5	6461	MFNX MFN - Metromedia Fiber Network	948	0, 41
6	286	KPN KPN Internet Backbone.	779	0, 45
7	2914	NTT-COMMUNICATIONS-2914 - NTT America, Inc.	774	0, 27
8	6453	GLOBEINTERNET TATA Communications	627	0, 27
9	577	BACOM - Bell Canada	523	0, 73
10	6762	SEABONE-NET TELECOM ITALIA SPARKLE S.p.A.	454	0,41

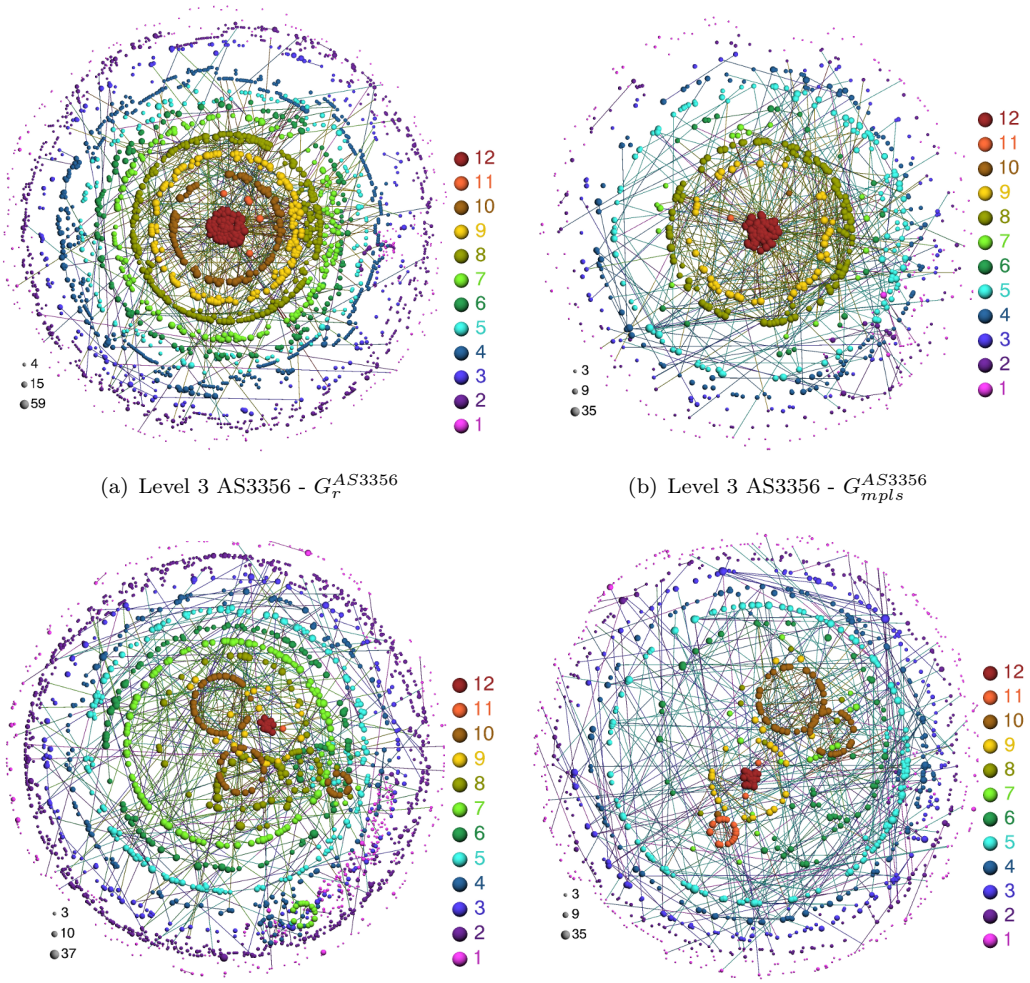
Tabla 3.4: Ranking *ASs* de acuerdo a la cantidad de enlaces *MPLS* intra-AS descubierta. Solamente se muestran los 10 primeros *ASs*. El campo fracción representa la relación entre los enlaces *MPLS* intra-AS respecto al total de enlaces encontrados.

Sistema Autónomo	G_r^{as}		G_{mpls}^{as}		$G_{r\setminus mpls}^{as}$	
	Vértices	**Aristas**	**Vértices**	**Aristas**	**Vértices**	**Aristas**
Level 3 AS3356	2913	12182	929	3692	1989	5104
AT&T AS7018	2230	6111	1265	3596	974	1212
LGI-UPC AS6830	967	2421	739	1530	240	303
BHN AS33363	248	1108	194	955	58	75

Tabla 3.5: Tamaño de los grafos que representan a los AS con mayor cantidad de enlaces *MPLS*.

estos grafos se describen en la tabla 3.5. En la figura se evidencia con más claridad lo que ya se había observado en la sección 3.7.1.2, que los dominios *MPLS* actúan casi como concentradores de *routers IP* de bajo grado, esto se evidencia en el hecho que los nodos de las capas periféricas se conecten con los nodos de mayor grado, nodos que aparecieron solamente después del proceso de contracción *MPLS* y que por tanto corresponden a dominios *MPLS*.

De la figura 3.17 también se puede obtener una idea de la cantidad de dominios *MPLS*



(a) Level 3 AS3356 - G_r^{AS3356}

(b) Level 3 AS3356 - G_{mpls}^{AS3356}

(c) AT&T AS 7018 - G_r^{AS7018}

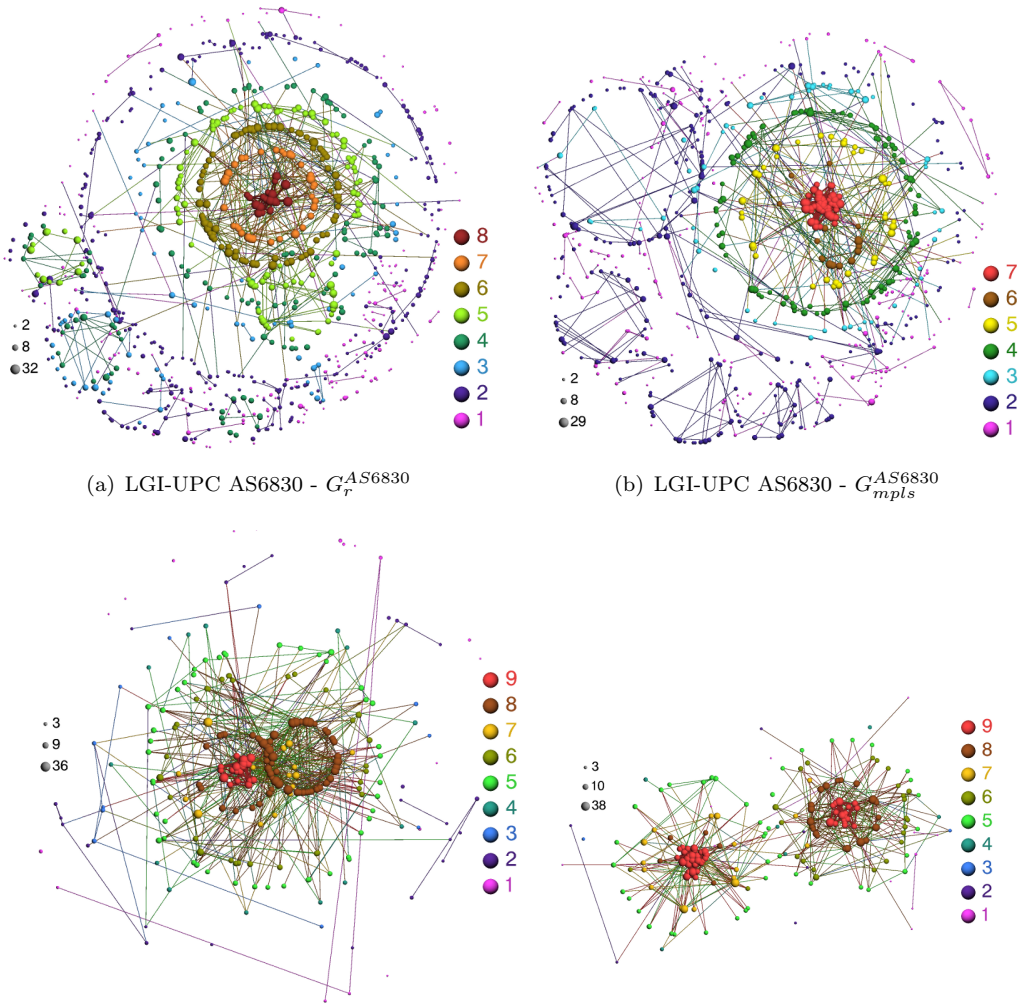
(d) AT&T AS 7018 - G_{mpls}^{AS7018}

FIGURA 3.15: Visualizaciones basadas en K-núcleos de los grafos G_r^{as} y G_{mpls}^{as} para los sistemas autónomos AS3356 y AS7018.

que se encuentran dentro de un mismo *AS*. Se observa que en todos los *AS* hay al menos un dominio *MPLS* bien definido que se destaca por su alto grado respecto al resto de nodos, es decir, todo el Dominio *MPLS* se contrajo a un nodo de alto grado, ubicado típicamente en las capas centrales de las distintas figuras analizadas.

3.7.2. Enlaces *MPLS* inter-*AS*

Previamente se describieron algunos aspectos del comportamiento de los enlaces *MPLS* al interior de un *AS*. Si bien la mayoría de enlaces *MPLS* descubiertos se encontraban al interior del *AS*, es importante evaluar los túneles *MPLS* de un *AS* hacia otro. En la presente sección se analiza la frecuencia con la que los enlaces *MPLS* aparecen entre *ASs* distintos, las características del grafo a nivel de *ASs* considerando solamente las



(a) LGI-UPC AS6830 - G_r^{AS6830}

(b) LGI-UPC AS6830 - G_{mpls}^{AS6830}

(c) BRIGHT HOUSE NETWORKS AS33363 - $G_r^{AS33363}$

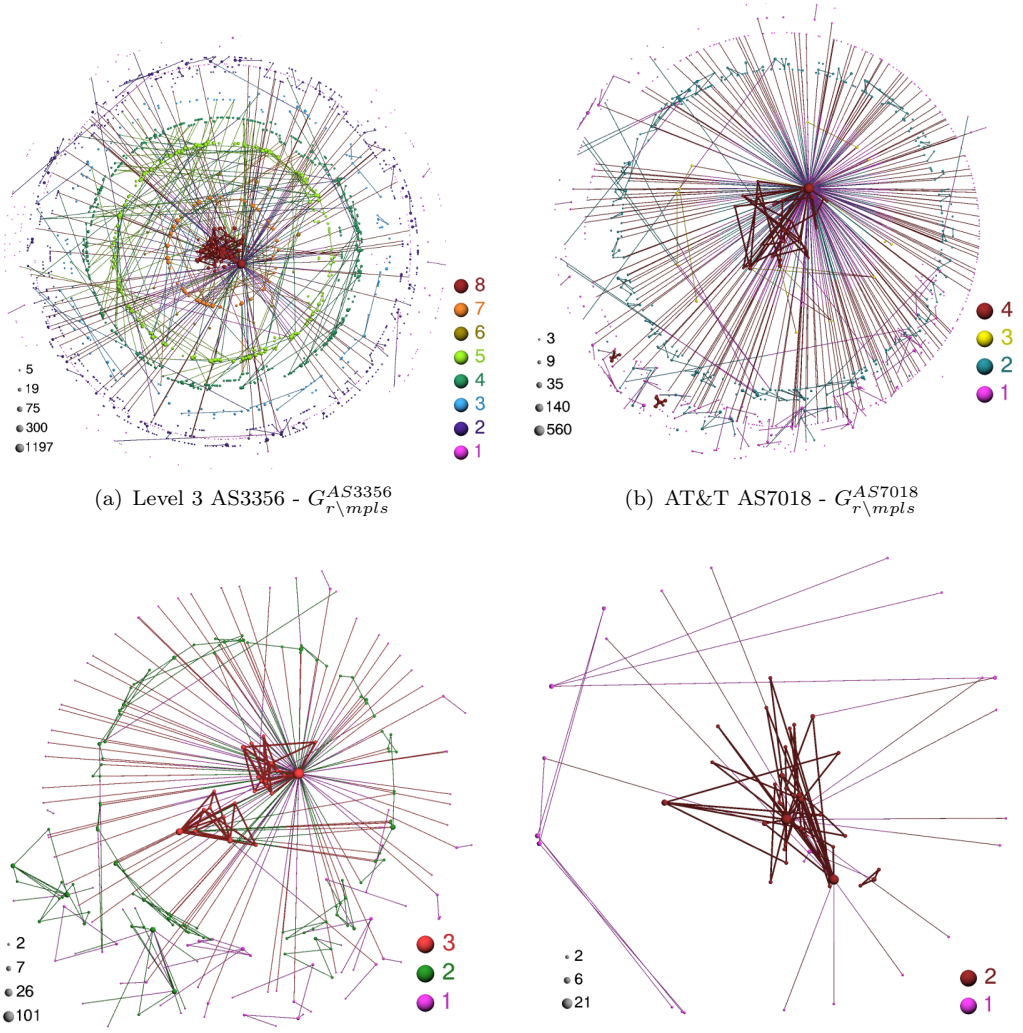
(d) BRIGHT HOUSE NETWORKS AS33363 - $G_{mpls}^{AS33363}$

FIGURA 3.16: Visualizaciones basadas en K-núcleos de los grafos G_r^{as} y G_{mpls}^{as} para los sistemas autónomos AS6830 y AS33363.

vecindades alcanzadas mediante túneles *MPLS* y finalmente se analiza la relación que existe entre los enlaces *MPLS* inter-*AS* e intra-*AS*.

La figura 3.18 muestra un histograma acumulativo de la cantidad de vecinos que un *AS* alcanza mediante túneles *MPLS*. Se observa que la mayoría de los sistemas autónomos tienen escasos enlaces inter-*ASs*: el 82 % de *AS* tienen al menos 4 sistemas autónomos como vecinos alcanzados mediante enlaces *AS*. Sin embargo existen también *AS* que alcanzan hasta 38 sistemas autónomos usando enlaces *MPLS*.

Para obtener más información del comportamiento de las conexiones entre *AS* mediante enlaces *MPLS* se analizó el grafo $G_{mpls}^{inter-as} = (V_{as}, E_{mpls}^{inter-as})$. Los vértices corresponden



(a) Level 3 AS3356 - $G_{r\backslash mpls}^{AS3356}$

(b) AT&T AS7018 - $G_{r\backslash mpls}^{AS7018}$

(c) LGI-UPC Liberty Global Operations AS6830- $G_{r\backslash mpls}^{AS6830}$

(d) BRIGHT HOUSE NETWORKS LLC AS33363- $G_{r\backslash mpls}^{AS33363}$

FIGURA 3.17: Visualizaciones basadas en K-núcleos del grafo $G_{r\backslash mpls}^{as}$.

a *ASs* y las aristas a enlaces *MPLS* entre *ASs*. Los detalles del tamaño de este grafo se muestran en la tabla 3.3.

En la figura 3.19 se muestra la Distribución de Probabilidad de grados del grafo $G_{mpls}^{inter-as}$, se observa que la distribución puede aproximarse a una ley de potencias conforme a la ecuación 2.2, donde $\gamma = 2,41$.

La figura 3.20 muestra el grado medio de los vecinos del grafo $G_{mpls}^{inter-as}$, se observa que el grado medio de los vecinos disminuye cuando aumenta el grado del *AS*, presentando una ligera tendencia discordante. Es decir, los *ASs* de menor grado se conectan con los *ASs* de mayor grado, quienes actúan como concentradores.

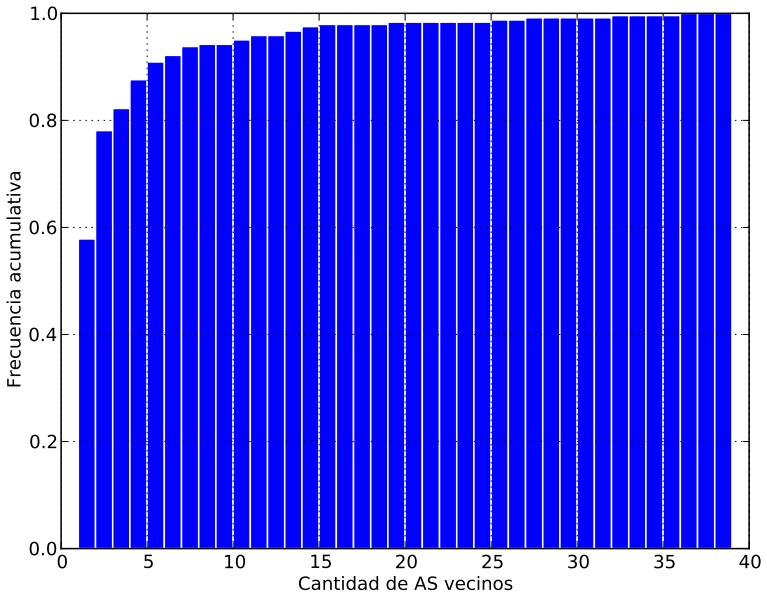


FIGURA 3.18: La figura superior representa un histograma acumulativo de la cantidad de vecinos que tiene un *AS* a través de enlaces *MPLS*. El eje horizontal representa el número de vecinos n que tiene un *AS* y el eje vertical representa la frecuencia con la que se encontró al menos n *AS* como vecinos mediante enlaces *MPLS*. En el 82 % de observaciones, los *AS* tienen como vecinos menos de 4 sistemas autónomos conectados mediante túneles *MPLS*.

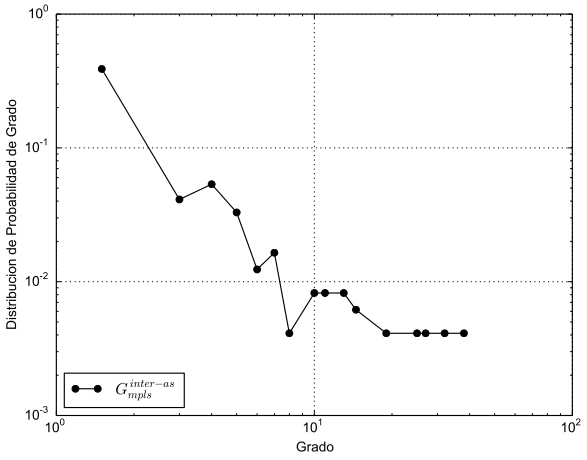


FIGURA 3.19: Distribución de Probabilidad de Grado de $G_{mpls}^{inter-as}$.

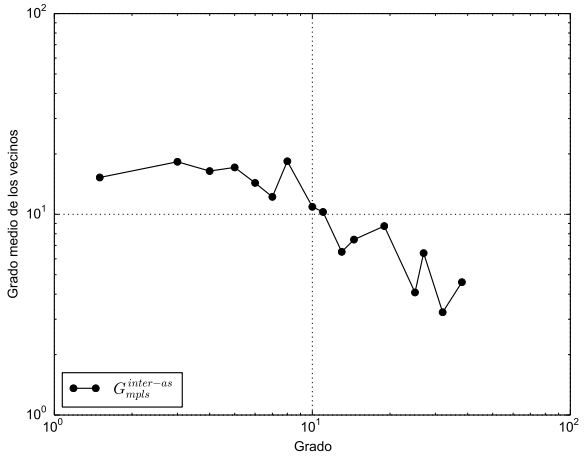


FIGURA 3.20: Grado medio de los Vecinos de $G_{mpls}^{inter-as}$.

La figura 3.21 muestra el coeficiente de *Clustering* del grafo $G_{mpls}^{inter-as}$, el cual se mantiene constante con excepción de los *ASs* con alto grado en donde disminuye. Es decir, los nodos vecinos de los nodos de menor grado están más conectados entre sí, que los vecinos de los nodos de grado elevado. Esta observación es consecuente con el comportamiento observado en el Grado medio de los vecinos.

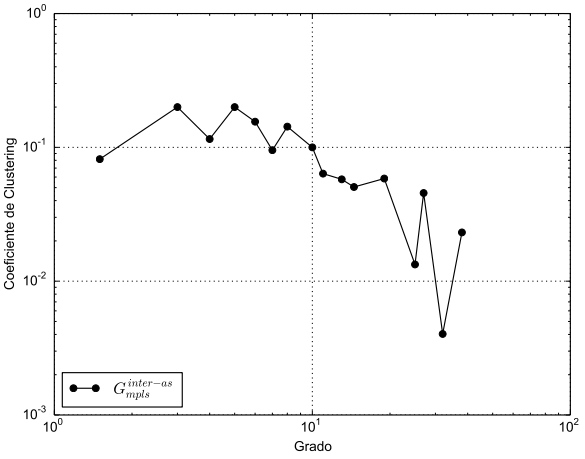


FIGURA 3.21: Coeficiente de Clustering de $G_{mpls}^{inter-as}$.

En la Figura 3.22 se muestra la visualización del grafo basada en *K-núcleos*. En la figura se observa que la mayoría de los nodos ubicados en las *capas* periféricas se conectan directamente con el núcleo central, siendo poco frecuentes las conexiones entre la periferia y las capas intermedias. Esta observación permite concluir que los nodos ubicados en

el núcleo central actúan como concentradores. Otra evidencia de esto es que a pesar de tener nodos con grado hasta de 40, la máxima capa es 4 y pocos nodos se ubican en las capas intermedias. El comportamiento del grado medio de los vecinos observado en la figura 3.20 también corrobora esta conclusión.

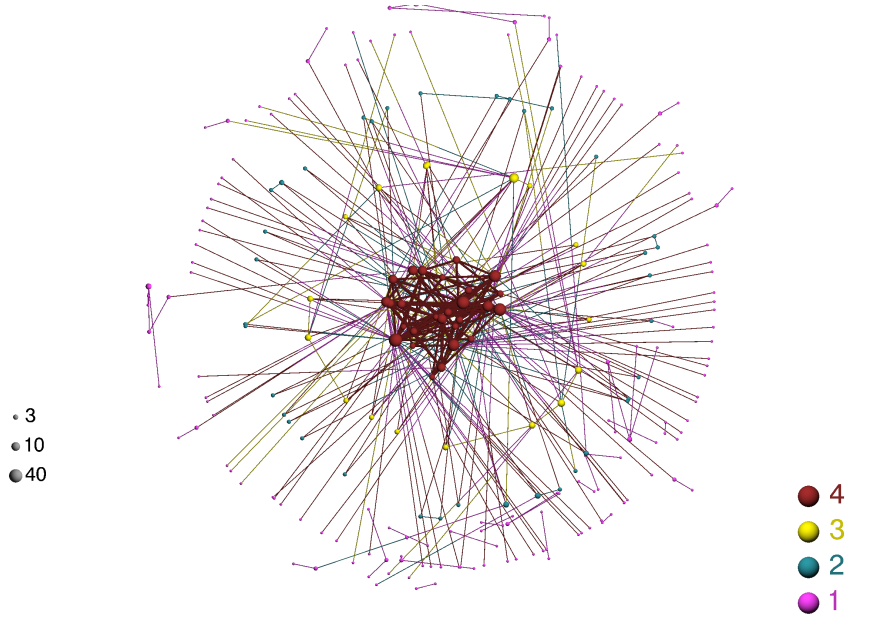


FIGURA 3.22: Visualización basada en *k-núcleos* de $G_{mpls}^{inter-as}$.

Más detalles de los *ASs* ubicados en el núcleo central se muestran en la tabla 3.6. La lista corresponde a los 10 *ASs* con mayor grado, pero como se mencionó previamente, los *ASs* de mayor grado coinciden con aquellos que se ubican en el núcleo central. Como dato adicional se muestra el número de enlaces inter-*ASs* observados. Se observa que el número de vecinos es menor al número de enlaces *MPLS* lo que significa que dos sistemas autónomos se conectan usando más de un enlace *MPLS*.

Finalmente, se analizó la relación entre los *AS* con mayor cantidad de enlaces *MPLS* intra-*AS* (tabla 3.4) y aquellos con mayor cantidad de enlaces *MPLS* inter-*AS* (tabla 3.6). Con esta información se analiza la relación entre los enlaces *MPLS* dentro de un *AS* y los enlaces *MPLS* dirigidos a otros *ASs*. De esta comparación se puede concluir que aparentemente la presencia de enlaces *MPLS* intra-AS no involucra tener enlaces *MPLS* inter-AS y viceversa. Solamente 3 sistemas autónomos aparecen en las dos tablas (AS3356, AS6762, AS6453). Sin embargo en aquellos *ASs* que aparecen ranqueados por el

	Número de AS	**Nombre de AS**	**Vecinos alcanzados mediante túneles *MPLS***	**Enlaces inter-AS**
1	3356	LEVEL3 Level 3 Communications	40	247
2	701	UUNET - MCI Communications Services - Verizon	36	165
3	24709	MNI MNI Telecom S.A. IP Backbone,PL	32	96
4	3257	TINET-BACKBONE Tinet SpA	27	174
5	1200	AMS-IX1 Amsterdam Internet Exchange B.V.	25	70
6	6762	SEABONE-NET TELECOM ITALIA SPARKLE S.p.A.	19	117
7	12389	ROSTELECOM-AS OJSC Rostelecom	15	157
8	1299	TELIANET TeliaSonera International Carrier	14	52
9	6453	GLOBEINTERNET TATA Communications	14	122
10	3549	GBLX Global Crossing Ltd	13	46

Tabla 3.6: Ranking de *ASs* con mayor cantidad de vecinos mediante túneles *MPLS*. Solamente se muestran los 10 primeros *ASs*.

número de enlaces *MPLS* inter-AS pero no por enlaces *MPLS* intra-AS, se encontró que existe un gran porcentaje de túneles implícitos, como se muestra en la tabla 3.7.

El hecho de que en los *ASs* mostrados en la tabla 3.7 se destaque la existencia de túneles inter-AS y que además los escasos enlaces *MPLS* intra-*AS* sean implícitos, cuando el comportamiento general es el contrario, es decir se tiene pocos enlaces intra-AS y pocos túneles implícitos, permite sospechar que se debe a que una gran cantidad de túneles *MPLS* no se están descubriendo. En estos *ASs* aunque no se puede cuantificar, se podría asegurar la existencia de túneles *MPLS* invisibles.

Número AS	**Nombre AS**	**% de túneles implícitos**
AS701	UUNET - MCI Communications Services - Verizon	89,04
AS24709	MNI MNI Telecom S.A. IP Backbone,PL	Sin túneles
AS3257	TINET-BACKBONE Tinet SpA	100,00
AS1200	AMS-IX1 Amsterdam Internet Exchange B.V.	100,00
AS12389	ROSTELECOM-AS OJSC Rostelecom	100,00
AS1299	TELIANET TeliaSonera International Carrier	25,66
AS3549	GBLX Global Crossing Ltd	98,98

Tabla 3.7: Porcentaje de túneles implícitos descubiertos en los sistemas autónomos con mayor cantidad de enlaces *MPLS* intra-*AS*.

Capítulo 4

Conclusiones

4.1. Conclusiones

En el presente trabajo se analizó el porcentaje de túneles *MPLS*, se puso a prueba la validez de los distintos métodos para descubrir un nodo *MPLS*, se analizaron las propiedades del grafo de Internet que se ven alteradas por la existencia de estos túneles y finalmente se encontraron algunos indicios novedosos para el descubrimiento de los túneles *MPLS* invisibles. Estas conclusiones se sintetizan en los siguientes puntos:

i En las exploraciones realizadas se observaron una mayor cantidad de túneles *MPLS* que en trabajos previos como [DLMP12] y [SBE11] (en donde ya se habla sobre el crecimiento de túneles *MPLS* año tras año). En los trabajos anteriores se encontró hasta un 30 % de túneles *MPLS* del total de exploraciones realizadas, mientras que en el presente trabajo se encontraron túneles *MPLS* en cerca del 40 % de las exploraciones (ver figura 3.14).

ii Se analizó la firma *u-turn* propuesta en [DLMP12], y se encontró que el uso de esta firma no permite predecir la posición de un *LSR* dentro del túnel, haciendo imposible verificar su confiabilidad (ver figuras 3.6 y 3.7).

iii El proceso de contracción de nodos *MPLS* no conlleva cambios en la distribución de probabilidad de grados pero sí implica un cambio en la distribución del grado medio de los vecinos (ver figura 3.8 y figura 3.9). Por otro lado, la distribución del coeficiente de *clustering* se ve afectada solamente por el proceso de resolución de

routers, pero no se ve afectada por el proceso de contracción de nodos *MPLS* (ver figura 3.10).

iv Al analizar las visualizaciones basadas en *k-núcleos* de los grafos completos de Internet: G_{ip}, G_r y $G_{r\backslash mpls}$, se observó que después del proceso de contracción de nodos *MPLS*, en el grafo $G_{r\backslash mpls}$ aparecen nodos de grado alto en capas bajas o periféricas (ver figura 3.13). Por otro lado, en las visualizaciones basadas en *k-núcleos* para los grafos $G_{ip}, G_r, G_{r\backslash mpls}$ (ver figuras 3.11, 3.12 y 3.13), aparecen componentes desconexas cuyo origen no está claro, sospechando que pueden originarse debido a secciones *IPv6* que no fueron descubiertas en la exploración o a fenómenos relacionados con el *512K day*[1].

v Los análisis de los grafos formados por enlaces pertenecientes a un mismo *AS*, muestran que en los grandes *ASs*, más del 30 % de los enlaces corresponden a enlaces *MPLS* (ver tabla 3.4). Respecto a las visualizaciones basadas en *k-núcleos* de estos mismos grafos, se observa que no existen cambios en su estructura: Los grafos G_r^{as} y G_{ip}^{as} son bastante similares en cuando a su estructura de *k-núcleos*, a lo sumo aparecen capas menos pobladas (ver figuras 3.15 y 3.16). En cambio por otro lado, el proceso de contracción *MPLS* modifica drásticamente la estructura de *k-núcleos*, disminuyendo el número de capas y aumentando el grado de los nodos del grafo $G_{r\backslash mpls}^{as}$ (ver figura 3.17).

vi Finalmente se observó que los enlaces *MPLS* entre distintos *ASs* no son frecuentes y que la mayoría de estos enlaces están concentrados en un pequeño número de nodos (ver figura 3.22). Así mismo se encontró que la existencia de un elevado porcentaje de túneles *MPLS* implícitos dentro de un mismo *AS* está asociada a que el *AS* tenga más enlaces *MPLS* hacia el exterior que hacia sí mismo (ver tablas 3.6 y 3.7). Este comportamiento no obedece las observaciones generales, en donde los túneles invisibles no representan más que el 10 % y en donde lo más común es que existan más enlaces *MPLS* dentro de un mismo *AS*, lo cual sería un indicio de la existencia de túneles invisibles.

[1] http://research.dyn.com/2014/08/Internet-512k-global-routes/ Título no oficial al evento en el cual varios *routers* de Internet alcanzaron el límite de 512K rutas *IPv4 BGP*, causando varios cortes en *Internet*

El trabajo realizado permite comprender la importancia y el impacto de los túneles *MPLS* sobre Internet, encontrando un alto volumen de penetración, que analizado conjuntamente con la cantidad de túneles encontrados en trabajos anteriores, se estima continuarán en crecimiento. Por otro lado, los cambios que los túneles *MPLS* producen sobre las propiedades del grafo de Internet, demuestran la importancia de estudiar los túneles para conocer con mayor precisión y detalle la topología de Internet. Asimismo, se cree que en los estudios posteriores de los túneles *MPLS*, se debe asignar especial importancia a los túneles invisibles y por otro lado, ignorar los túneles implícitos detectados usando la firma *u-turn*, ya que está asociada a errores causados por el balanceo de carga existente en Internet.

Bibliografía

[AFT07] Brice Augustin, Timur Friedman y Renata Teixeira. «Measuring Load-balanced Paths in the Internet». En: *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. IMC '07. San Diego, California, USA: ACM, 2007, págs. 149-160. URL: http://doi.acm.org/10.1145/1298306.1298329.

[AH06] José Ignacio Alvarez-Hamlein. «Taxonomía de los modelos de Topología de Internet». En: *Mecánica Computacional* 25 (2006), págs. 2597-2612.

[AHBV06] J. Ignacio Alvarez-Hamelin, Alain Barrat y Alessandro Vespignani. «Large scale networks fingerprinting and visualization using the k-core decomposition». En: *Advances in Neural Information Processing Systems 18*. MIT Press, 2006, págs. 41-50.

[Aug+06] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien y Renata Teixeira. «Avoiding Traceroute Anomalies with Paris Traceroute». En: *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. IMC '06. Rio de Janeriro, Brazil: ACM, 2006, págs. 153-158. URL: http://doi.acm.org/10.1145/1177080.1177100.

[BGTP07] R. Bonica, D. Gan, D. Tappan y C. Pignataro. *ICMP Extensions for Multiprotocol Label Switching*. RFC 4950. Internet Engineering Task Force, ago. de 2007. URL: http://www.rfc-editor.org/rfc/rfc4950.txt.

[BSK06] Sevcan Bilir, Kamil Saraç y Turgay Korkmaz. «Intersection Characteristics of End-to-End Internet Paths and Trees.» En: *ICNP*. IEEE Computer Society, 31 de ene. de 2006, págs. 378-390. ISBN: 0-7695-2437-0. URL: http://dblp.uni-trier.de/db/conf/icnp/icnp2005.html#BilirSK05.

[Cal+97] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow y A. Viswanathan. En: (mayo de 1997). URL: http://tools.ietf.org/html/draft-ietf-mpls-framework-00.

[CIS] CISCO. *How does load balancing work?* URL: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml.

[CSN09] Aaron Clauset, Cosma Rohilla Shalizi y Mark E. J. Newman. «Power-Law Distributions in Empirical Data». En: *SIAM Review* 51.4 (2009), págs. 661-703. DOI: http://dx.doi.org/10.1137/070710111. URL: \url{http://arxiv.org/abs/0706.1062}.

[DB+07] Giuseppe Di Battista, Thomas Erlebach, Alexander Hall, Maurizio Patrignani, Maurizio Pizzonia y Thomas Schank. «Computing the types of the relationships between autonomous systems». En: *IEEE/ACM Trans. Netw.* 15.2 (abr. de 2007), págs. 267-280. ISSN: 1063-6692. URL: http://dx.doi.org/10.1109/TNET.2007.892878.

[Dim+07a] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k claffy y G. Riley. «AS Relationships: Inference and Validation». En: *ACM SIGCOMM Computer Communication Review (CCR)* 37.1 (ene. de 2007), págs. 29-40.

[Dim+07b] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy y G. Riley. «AS Relationships: Inference and Validation». En: *ACM SIGCOMM Computer Communication Review (CCR)* 37.1 (ene. de 2007), págs. 29-40.

[DLMP12] B. Donnet, M. Luckie, P. Mérindol y J. Pansiot. «Revealing MPLS tunnels obscured from traceroute». En: *ACM SIGCOMM Computer Communication Review (CCR)* 42.2 (abr. de 2012), págs. 87-93.

[Don13] Benoit Donnet. *Internet Topology Discovery*. Springer Berlin Heidelberg, ene. de 2013. URL: http://hdl.handle.net/2268/145260.

[Gao01] Lixin Gao. «On inferring autonomous system relationships in the internet». En: *IEEE/ACM Trans. Netw.* 9.6 (dic. de 2001), págs. 733-745. URL: http://dx.doi.org/10.1109/90.974527.

[GS09] Mehmet H. Gunes y Kamil Sarac. «Resolving IP Aliases in Building Traceroute-based Internet Maps». En: *IEEE/ACM Trans. Netw.* 17.6 (dic. de 2009), págs. 1738-1751. ISSN: 1063-6692. URL: http://dx.doi.org/10.1109/TNET.2009.2014227.

[HFc12] Bradley Huffaker, Marina Fomenkov y kc claffy. «Internet Topology Data Comparison». En: *CAIDA Technical Report* (mayo de 2012). URL: http://www.caida.org/research/topology/topo_comparison/.

[JUN] JUNIPER. *Configuring Per-Packet Load Balancing*. URL: http://www.juniper.net/techpubs/en_US/junos12.2/topics/usage-guidelines/policy-configuring-per-packet-load-balancing.html.

[Mar+11] Pietro Marchetta, Pascal Mérindol, Benoit Donnet, Antonio Pescapé y Jean-Jacques Pansiot. «Topology Discovery at the Router Level: A New Hybrid Tool Targeting ISP Networks». En: *IEEE Journal on Selected Areas in Communication, Special Issue on Measurement of Internet Topologies* (2011).

[Mar+12] Pietro Marchetta, Pascal Mérindol, Benoit Donnet, Antonio Pescapé y Jean-Jacques Pansiot. «Quantifying and mitigating IGMP filtering in topology discovery.» En: *IEEE Global Communications Conference (GLOBECOM)*. (Dic. de 2012).

[MDBP10] Pascal Mérindol, Benoit Donnet, Olivier Bonaventure y Jean-Jacques Pansiot. «On the impact of layer-2 on node degree distribution». En: IMC '10 (2010), págs. 179-191. URL: http://doi.acm.org/10.1145/1879141.1879164.

[MM00] K. Muthukrishnan y A. Malis. *A Core MPLS IP VPN Architecture*. RFC 2917. Internet Engineering Task Force, sep. de 2000. URL: http://www.rfc-editor.org/rfc/rfc2917.txt.

[Mé+09] Pascal Mérindol, Virginie Van den Schrieck, Benoit Donnet, Olivier Bonaventure y Jean-Jacques Pansiot. «Quantifying ASes Multiconnectivity Using Multicast Information». En: *Proc. ACM USENIX Internet Measurement Conference (IMC)* (nov. de 2009). URL: http://inl.info.ucl.ac.be/publications/quantifying-ases-multiconnectivity-using-multicast-information.

[New02] M. E. J. Newman. «Assortative Mixing in Networks». En: *Physical Review Letters* 89.20 (oct. de 2002), pág. 208701. DOI: 10.1103/PhysRevLett.89.208701. URL: http://link.aps.org/doi/10.1103/PhysRevLett.89.208701.

[Pos81] J. Postel. *Internet Control Message Protocol*. RFC 3812 (Proposed Standard). United States, 1981.

[PSV04] R. Pastor-Satorras y A. Vespignani. *Evolution and Structure of Internet: A Statistical Physics Approach*. Cambridge University Press, 2004. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521826983.

[RL95] Y. Rekhter y T. Li. *A Border Gateway Protocol 4 (BGP-4)*. RFC 1771. Internet Engineering Task Force, mar. de 1995. URL: http://www.rfc-editor.org/rfc/rfc1771.txt.

[RVC01] E. Rosen, A. Viswanathan y R. Callon. *Multiprotocol Label Switching Architecture*. RFC 3031. Internet Engineering Task Force, ene. de 2001. URL: http://www.rfc-editor.org/rfc/rfc3031.txt.

[SBE11] Joel Sommers, Paul Barford y Brian Eriksson. «On the Prevalence and Characteristics of MPLS Deployments in the Open Internet». En: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. IMC '11. Berlin, Germany: ACM, 2011, págs. 445-462. URL: http://doi.acm.org/10.1145/2068816.2068858.

[SVN04] C. Srinivasan, A. Viswanathan y T. Nadeau. *Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)*. RFC 3812. Internet Engineering Task Force, jun. de 2004. URL: http://www.rfc-editor.org/rfc/rfc3812.txt.

[TGSE01] Hongsuda Tangmunarunkit, Ramesh Govindan, Scott Shenker y Deborah Estrin. «The Impact of Routing Policy on Internet Paths». En: *Proc. IEEE INFOCOM* (ene. de 2001).

[Tha00] D. Thaler. *Multipath issues in unicast and multicast next-hop selection. Internet Engineering Task Force: RFC 2991*. 2000.

[VPMD13] Yves Vanaubel, Jean-Jacques Pansiot, Pascal Mérindol y Benoit Donnet. «Network Fingerprinting: TTL-Based Router Signature». En: *ACM/USENIX Internet Measurement Conference (IMC)*. Barcelona, Spain, oct. de 2013.

[XG04] Jianhong Xia y Lixin Gao. «On the Evaluation of AS Relationship Inferences». En: *Proc. IEEE Global Communications Conference (GLOBECOM)* (nov. de 2004).