Técnicas de descubrimiento de topologías en Internet

Facultad de Ingeniería, Universidad de la República

Diego Kiedanski
Tutor: Eduardo Grampín

27 de noviembre de 2017

Quiero dedicar este trabajo a Debi, que me enseñó a reir más y mejor.

Agradecimientos

A mis viejos que no me dejaron abandonar esta carrera cuando no aguantaba más.

A la banda con la que compartí noches en el 401 y a la que no tenía nada que ver con la FING y me permitía cambiar el chip cada tanto. Va, y a los que tenían que ver con la FING, no compartimos 401 pero salíamos y hacían la vida más llevadera, también.

A Juliana por la clase de geometría hiperbólica.

A Eduardo, que no deber poder esperar a que termine este proyecto para liberarse de mi y me aguantó con infinita paciencia.

A Ignacio, que nos dio tremenda mano desde la UBA.

A LACNIC, que nos donó muchísimos créditos en Ripe Atlas sin los cuales no habríamos podido realizar la última campaña.

Índice general

Capítulo 1

Introducción

Internet, la red de las redes, está compuesta por ordenadores, routers, switches, cables y otros muchos dispositivos. Dado que se trata de una red tan compleja, en general al pensar en ella realizamos abstracciones. Podemos pensar en Internet a nivel de ordenadores personales, a nivel de direcciones IP, de proveedores de Internet como Antel o incluso países.

No es difícil encontrar argumentos sobre por qué debemos entender cómo funciona Internet: prácticamente nuestra vida gira alrededor de ella y es indispensable para poder mantener nuestro estilo de vida.

Increíblemente, la red de las redes, de las más grandes y complejas con las que interactúa el hombre, no cuenta con una organización central que la regule.

Esta característica que la hace sumamente independiente presenta un gran problema: no existe una fuente fiable con información de todo Internet. Lo que es más, los actores involucrados (proveedores de Internet, entre otros) tienen muchas veces intereses económicos en no revelar información sobre como se han conectado, con cuanto equipamiento cuentan, como se distribuye el mismo, etc.

En vista de todo esto, la tarea de *medir Internet* para crear un mapa preciso cobra un importancia aún mayor, no existen otras alternativas.

Dado que experimentar con Internet es costoso (imagínese probar un sustituto de BGP a nivel global), se hace necesario contar con modelos realistas. La forma natural de modelar Internet es mediante grafos. Pero no todos los grafos son buenas representaciones de Internet. Esto naturalmente lleva a la pregunta: ¿qué hace que un grafo sea un buen modelo de Internet?

Está pregunta es difícil de contestar y no tiene respuesta única. Lo que sí

sabemos responder es cuando un modelo es malo. Un modelo es malo si no presenta algunas de las propiedades que Internet sí (se detallaran al llegar a la sección Modelos).

¿Cómo se sabe que propiedades tiene Internet?

El camino es experimental y consiste en: medir Internet, generar una topología (grafo) a partir de lo medido y cuantificar las propiedades del grafo resultante. Esto, claro está, tiene muchos errores de medición, por lo que de por sí no es suficiente.

Desde hace años que se generan topologías de Internet y se ha encontrando que algunas métricas permanecen invariantes en el tiempo, lo que incita a pensar que son intrínsecas de Internet.

El auge en el área data de 1999, año en que los hermanos Faloutsos [14] encontraron que en topologías inferidas a partir de trazas BGP, la distribución de grado presentaba una forma particular, distinta a la observada en grafos aleatorios.

¿No podríamos conformarnos con el hecho de que Internet funciona? La respuesta, lamentablemente, es que muchas de las tecnologías actuales no están a la altura de los requisitos que imponemos sobre Internet. Hoy en día, una tabla de BGP tiene 750000 entradas sobre las que un router debe hacer fuerza bruta cada vez que quiere reenviar un paquete, el espacio de direcciones IPv4 está casi agotado, etc. Se ha argumentado también que IP no es adecuado para un Internet actual que se basa mayoritariamente en el acceso a contenido. La conclusión es que todavía falta mucho por hacer, y el primer paso en todas estas direcciones es medir.

En cuanto al objetivo de este trabajo es más bien de investigación, no se pretende obtener un producto final. En este sentido, esa tesis es más que nada el relato de un proceso: desde el estado del arte en su forma más teórica, hasta la puesta en práctica de lo aprendido.

En los capítulos dos y tres se resume el estado del arte en cuanto a técnicas de medición. Preguntas relevantes son: ¿cómo se mide?, ¿qué se mide? y ¿qué problemas pueden ocurrir midiendo?

Los capítulos cuatro versa sobre modelos. Se describen las principales propiedades a encontrar en los grafos y como distintos modelos producen resultados variados.

El capítulo cinco resume algunos proyectos para medir Internet que se diferencian de otros por su calidad (superior).

El capítulo seis describe un primer intento de medir una red. En ese caso se usó un simulador de eventos discretos para poder contrastar con la realidad

subyacente.

Los capítulos siete y ocho describen dos campañas de medición realizadas para descubrir la topología de América Latina.

El último capítulo presenta algunas conclusiones generales del proyecto.

Feliz lectura.

Capítulo 2

Técnicas basadas en Traceroute

La técnica más utilizada para la construcción de topologías es utilizando la herramienta Traceroute desarrollada por Van Jacobson en el año 1987. En este capítulo se explicará como funciona, como se puede utilizar para inferir topologías y cuáles son algunos problemas que pueden llegar a ocurrir.

2.1. Generalidades

Traceroute es una herramienta que permite descubrir el camino entre dos nodos en Internet A y B.

Una secuencia de paquetes con TTL incrementados de forma gradual $(1, 2, \ldots, N)$ son enviados desde el nodo A al nodo B. Cuando un router recibe un paquete, decrementa su TTL en uno. Si el nuevo TTL es 0, envía un mensaje ICMP al nodo origen (A). De esta forma, A conoce una secuencia de interfaces en el camino A-B.

El proceso termina cuando se recibe un mensaje *Destino inalcanzable*, pues en general se envía un datagrama UDP con un puerto en el que no se espera que B esté escuchando.

Durante este proceso, se ejecuta un timer para cada paquete: en caso de no obtener una respuesta en un tiempo determinado (5 segundos por defecto), se asume que el router no contesta (*router anónimo*).

Por defecto se envían tres mensajes con el mismo TTL, es decir que si el nodo B se encuentra a N saltos de distancia se enviarán $3N$ paquetes.

Salto (TTL)	IP 1	IP 2	IP 3
1	192.168.1.1		
2	192.168.144.1		
3	81.46.72.233	80.58.81.165	81.46.72.237
4	81.46.8.94		
5	176.52.253.93		
6	5.53.1.74		
7	72.14.233.161	72.14.234.231	72.14.233.161
8	216.239.48.249	216.239.48.105	
9	8.8.8.8		

Cuadro 2.1: Traceroute desde 192.168.1.44 hacia 8.8.8.8

2.2. Infiriendo una topología

Un posible modelo es asociar direcciones IP a nodos. Dado el resultado de una ejecución de traceroute como se ve en el Cuadro 2.1, se puede inferir la topología observada en la Figura 2.1.

La IP desplegada en la salida es la que se encuentra en el encabezado de la respuesta. Dado que se envían 3 paquetes, es posible que estos tomen distintos caminos y a eso se debe la presencia de diversas direcciones IP en un mismo nivel. El otro posible escenario es que se tome el mismo camino pero el router responda con diversas interfaces y por lo tanto distintas direcciones IP. En la práctica se usa una sola IP por salto. Esa IP se elige basada en algún criterio que explicaremos mejor más adelante.

2.3. Primeras dificultades

En la práctica, se presentan diversas dificultades a la hora de utilizar Traceroute para inferir topologías.

Para empezar es posible que los paquetes se pierdan, en cuyo caso no se obtiene una respuesta. Este escenario en principio es simple de solucionar pues se podría intentar en otro momento en que la red esté menos congestionada.

Por otra parte, uno de los mayores problemas encontrados es la presencia de firewalls. En este sentido, los mismos pueden estar configurados para bloquear la entrada de datagramas UDP con puertos extraños (comporta-

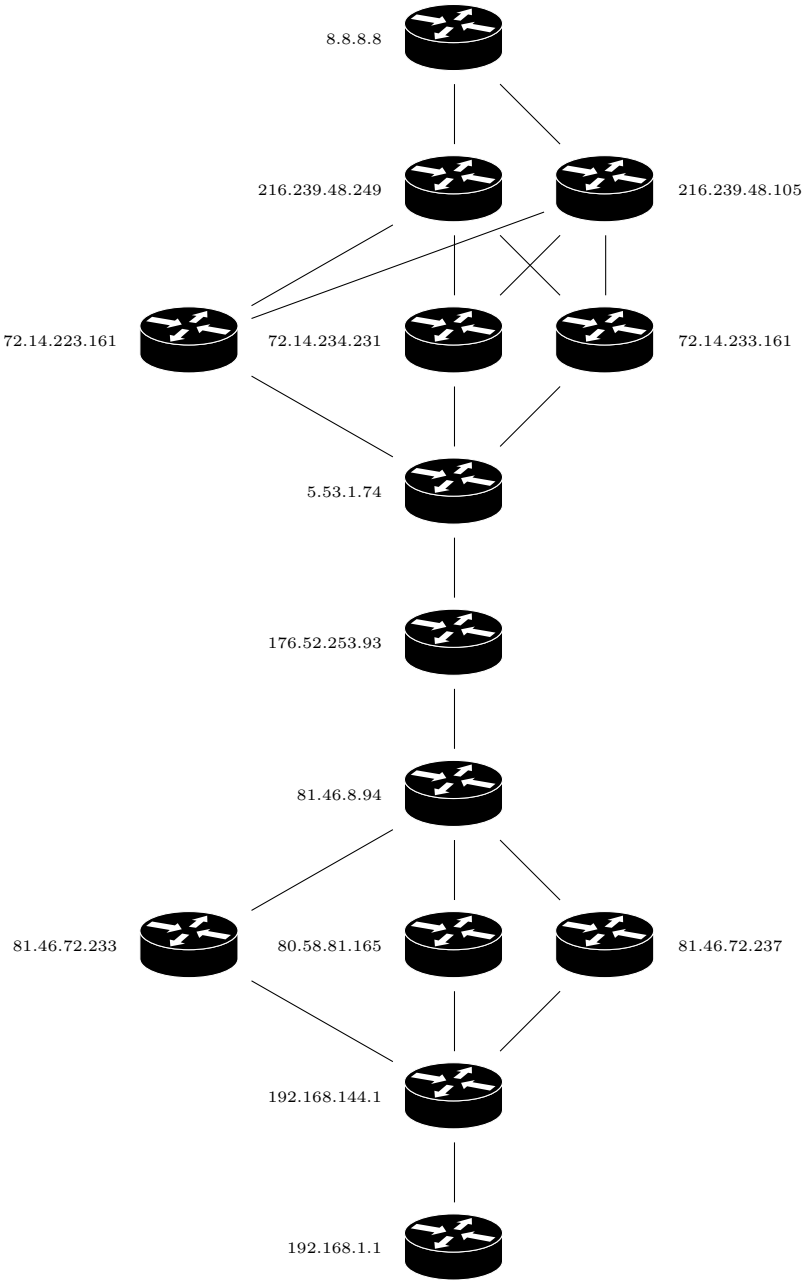


Figura 2.1: Topología inferida (*naive*) a partir del traceroute mostrado en el Cuadro 2.1

miento por defecto de traceroute). Para resolver el inconveniente anterior se han desarrollado variantes de traceroute en las cuales en lugar de enviar un datagrama UDP, se envía un paquete TCP SYN o un mensaje ICMP *Echo Request*. Para el segundo caso no existe una solución a priori. En la misma línea, podría pasar que el firewall esté configurado para evitar la salida de mensajes ICMP [13]. Dado que muchos proveedores no desean que la estructura interna de su red sea conocida, no se trata de un caso muy excepcional.

Otro inconveniente puede ocurrir en la presencia de un balanceador de carga en el camino hacia el destino (como L en la Figura 2.2 izquierda). En esta situación es posible que no se encuentren todos los caminos existentes. En la red presente en la Figura 2.2 la probabilidad de que uno de los dos caminos (A-C) o (B-D) no sea descubierto es 0,25. Para observar esto basta con observar que la probabilidad de que L enrute los tres paquetes enviados hacia Dst por A (o B de forma equivalente) es $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$ (asumiendo independencia).

2.4. Falsas inferencias

Dado que Traceroute fue pensada como una herramienta de diagnóstico y no de inferencia, es posible que aún cuando la ejecución termina de forma exitosa, la topología presente falsas aristas.

Un ejemplo típico se puede encontrar volviendo al escenario de la Figura 2.2 . El primer paquete lo recibe L con un $TTL = 1$. El segundo paquete es balanceado hacia A con un $TTL = 2$. El tercer paquete es balanceado hacia $B - D$ con un $TTL = 3$ y es D quien responde. Por último, el paquete llega a E con $TTL = 5$ y a Dst con $TTL = 6$. En este escenario, se infiere incorrectamente la arista $A - D$.

2.4.1. Loops

En [6] se discuten diversos escenarios que pueden derivar en la inferencia de loops. Esto ocurre cuando una misma dirección IP aparece de forma repetida como resultado de una ejecución. Está claro que la presencia de loops es un fenómeno que no ocurre en la realidad modelada y por lo tanto se debe corregir.

La presencia de un balanceador de carga y caminos de distinto largo es una posible explicación. Se observa en la Figura 2.3 que el nodo E puede

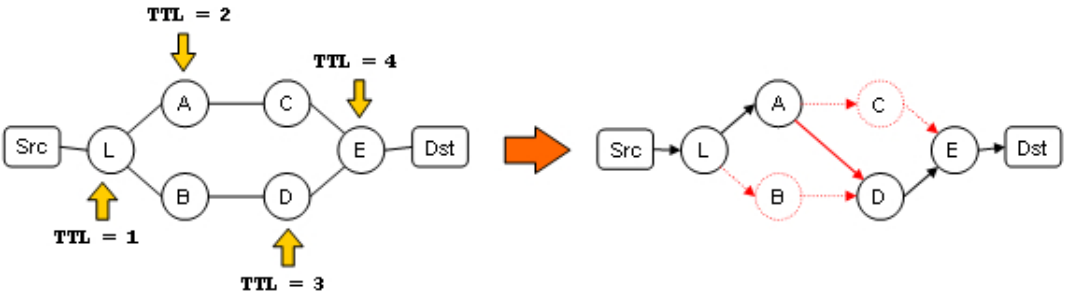


Figura 2.2: Ejemplo de un posible Traceroute frente a un balanceador de carga. Figura extraída de [54]

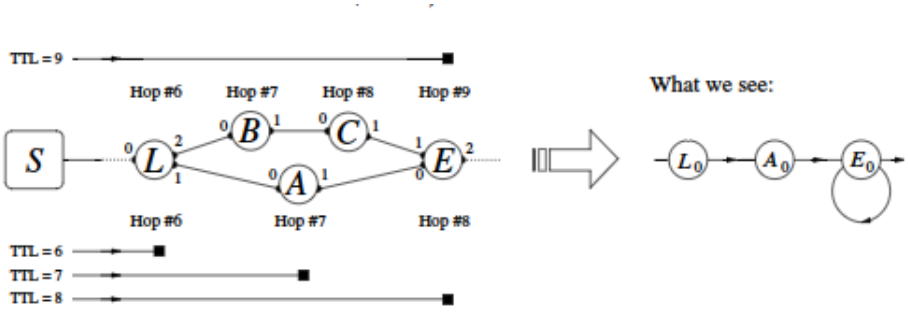


Figura 2.3: Balanceador de carga deriva en inferencia de loop. Figura extraída de [6]

responder a un paquete con $TTL = 8$ y a otro con $TTL = 9$ de tomarse los dos caminos.

Algunos routers como los que se encuentran en el borde de una NAT o algunos firewalls modifican la dirección de origen de los mensajes ICMP de forma que todos los routers en una misma red respondan con la misma IP. En la Figura 2.4 se ve como N,B y C se terminan infiriendo como un solo nodo con un loop.

Por último, un problema de configuración puede ocasionar que un router envíe un paquete con $TTL = 0$, forzando a que el router siguiente responda dos veces. En la figura 2.5 se ve como al F estar mal configurado, deriva en un loop en A.

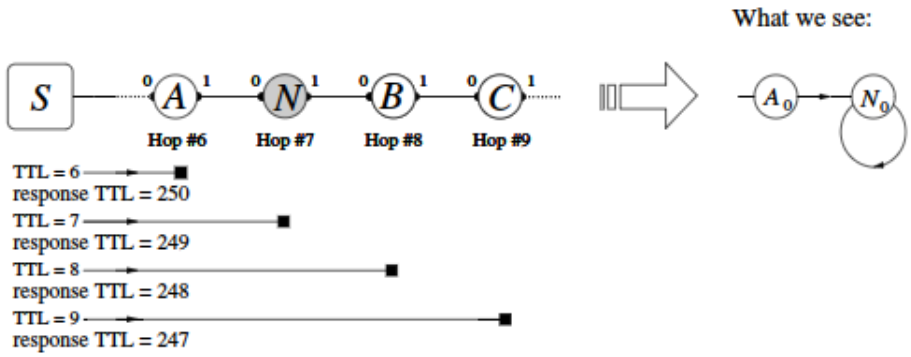


Figura 2.4: N, B y C se encuentran detrás de un NAT y responden con el mismo IP. Figura extraída de [6]

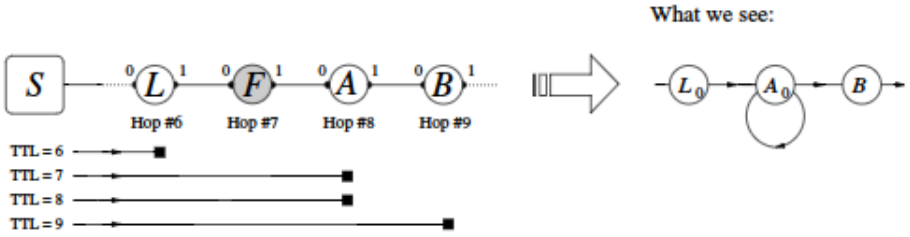


Figura 2.5: El router F incorrectamente reenvía un paquete con $TTL = 0$. Figura extraída de [6]

2.4.2. Paris-Traceroute

Para solucionar algunos de estos problemas, un nuevo programa fue desarrollado: Paris-Traceroute [54].

La implementación tradicional de traceroute de Jacobson utiliza el número de puerto para identificar los mensajes enviados con las respuestas, enviando cada nuevo paquete con un puerto distinto. Dado que el balanceo de carga muchas veces utiliza dicho campo para identificar flujos, para evitar esa situación se debe modificar el código fuente.

En lugar de modificar el puerto, Paris-Traceroute modifica campos en el encabezado de la capa de transporte. Específicamente: Checksum y Sequence Number para traceroutes usando UDP e ICMP (y TCP) respectivamente.

Cabe notar a pesar de estas modifiaciones, Paris-Traceroute no es capaz de mejorar el escenario en el que el balanceo ocurre por paquetes y no por flujos.

2.5. Resolución de Alias

Los mecanismos explicados hasta ahora permiten inferir una topología a nivel de intefaces. Sin embargo, es más natural y deseable estudiar la topología a nivel de routers. Para logar esto, fueron desarrolladas técnicas que permiten agregar distintas interfaces en un solo router. Este proceso se conoce como resolución de alias y una descripción más detallada puede encontrase en [[24], [18]].

2.5.1. Método basado en dirección

Este método descrito en [8], consiste en enviar un paquete a una dirección X con un puerto poco común. Si la respuesta es un mensaje ICMP: Port Unreachable con una dirección IP Y, entonces las direcciones X e Y pertenecen al mismo router.

La desventaja es que requiere que el router envíe mensajes ICMP.

2.5.2. Método basado en identificación

Inspirado en el campo Identificación, el método consiste en enviar un mensaje en las mismas condiciones que el método anterior a dos alias potenciales (sean x e y los identificadores de sus respectivas respuestas). Se envía

un tercer mensaje a la dirección que haya contestado antes y se guarda el identificador de su respuesta (z). Si $x < y < z$ y $z - x$ es pequeño (alrededor de 200), entonces se supone que las direcciones originales eran efectivamente alias.

Nuevamente este método no funciona si el router no envía mensajes ICMP. También puede fallar debido que el contador del identificador es cíclico y podrían darse números muy distintos, entre otras cosas.

2.5.3. Método basado en DNS

El mismo consiste en hacer una búsqueda reversa de DNS y obtener patrones comunes en los nombres obtenidos. A diferencia de los anteriores, no requiere que los routers respondan para poder identificarlos. Sin embargo, los nombres no siempre siguen una nomenclatura rigurosa por lo que puede resultar casi imposible para un programa resolver alias de esta forma (o hasta para un humano en caso de que no se siga ningún patrón).

2.5.4. Método basado en el grafo

Dado una serie de traceroutes, se puede construir un grafo G como fue explicado previamente en la subsección 2.2 donde los nodos son interfaces. Si se asume un esquema en el que las conexiones entre routers son punto a punto (no hay dispositivos de capa 2 de por medio), entonces una interfaz define un link (y por lo tanto un router). Asumiendo esto, se proponen dos reglas para descubrir alias:

- Si dos nodos A y B son predecesores a un nodo C, entonces A y B son alias.
- Si dos IPs aparecen en el mismo traceroute, entonces no pueden ser alias (asumiendo que no hay loops).

Un desarrollo más profundo así como métricas relacionadas a su performance pueden encontrarse en [50].

2.5.5. Método analítico

El método analítico pretende identificar dos alias basado en traceroutes simétricos (de ida y vuelta entre los mismos routers).

#	SMU a Yale	Yale a SMU
1	129.119.39.1	129.119.223.249
2	129.119.0.249	129.119.0.250
3	206.223.141.89	206.223.141.90
4	206.223.141.70	206.223.141.69
⋮	⋮	⋮

Cuadro 2.2: Traceroute entre SMU y Yale

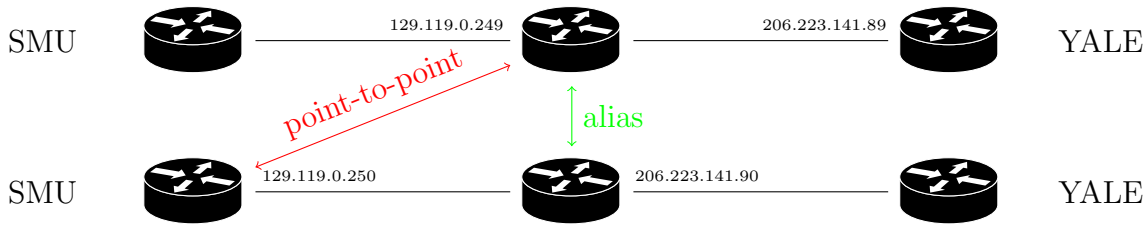


Figura 2.6: Fragmento de los traceroutes de ida y vuelta entre SMU y YALE para ilustrar como el método analítico resuelve alias

En general, a cada interfaz de un router se le asigna una subred distinta y en particular, para los enlaces punto a punto se utiliza un prefijo de 30 o 31 bits. Bajo esta hipótesis, dos direcciones IP consecutivas en caminos simétricos probablemente se correspondan con extremos de un enlace punto a punto.

Esta información se puede utilizar para inferir alias. Considérese el Cuadro 2.2.

Analizando la misma se puede ver que 129.119.0.249 y 129.119.0.250 forman un enlace punto a punto y por lo tanto 129.119.9.249 y 206.223.141.90 son alias. La Figura 2.6 ilustra la idea anterior.

La formalización, así como un algoritmo para este método pueden encontrarse en [18].

2.6. Resolución de routers anónimos

Otro problema a resolver al utilizar traceroute es la presencia de routers anónimos (indicados con un $*$ en la salida). Un mal manejo de los mismos puede derivar en topologías que poco tienen que ver con la realidad.

En [20], se presenta un algoritmo que utiliza la técnica: *inducción basada de grafos* (GBI) para determinar en ciertos casos si dos observaciones de routers anónimos ($*_a$ y $*_b$) se corresponden al mismo router o no.

2.7. Sesgo del muestreo

Uno de los mayores problemas que puede ocurrir cuando se realizan mediciones es el sesgo. Por sesgo nos podemos referir a dos cosas: errores consistentes en las mediciones realizadas, o la observación de tan solo algunos de los resultados posibles en el experimento.

No es difícil intuir que en una campaña de medición utilizando traceroute desde un único punto de sondeo, se descubrirán menos aristas que en una con varios puntos de sondeo. Ahora, si esta diferencia es solo cuantitativa, en principio podrían seguirse infiriendo propiedades generalizables del grafo. Por otra parte, si las diferencias son cualitativas, todo el proceso de inferencia se ve comprometido.

Lakhina et.al [26] sugirieron que la novedosa distribución de grado propuesta por los hermanos Faloutsos en [14], podría ser producto del sesgo en la medición y no una propiedad latente de Internet.

Con dicho fin, probaron que en un grafo Erdos-Reny (se definirá en la sección Modelos) disperso, la unión de los caminos más cortos de un pequeño subconjunto de los vértices (puntos de sondeo) a un conjunto mucho más grande de nodos (destinos), presenta una distribución que se asemeja a la descrita por Faloutsos. Esto implicaría que por más que el grafo original (topología real) no presenta dicha distribución, el resultado de la medición podría presentarla.

A partir del resultado anterior, muchos estudios continuaron esta línea de investigación, probando de forma más rigurosa el mismo resultado o extendiéndolo a otros tipos de grafos. Un ejemplo de estos trabajos es [1].

Otro resultado interesante propuesto en [26] se presenta en la forma de test de hipótesis con los cuales corroborar si un dataset dado presenta o no sesgo.

2.8. Resolución de ASes

Muchas veces la topología que se quiere inferir es a nivel de sistemas autónomos y no de routers. En estos casos, también es posible utilizar técnicas basadas en traceroute. Así como dadas muchas interfaces se puede resolver un router, también se puede resolver un AS.

Esto se puede realizar mirando tablas de BGP y asociando todas las IP dentro de un prefijo con el primer ASN encontrado en el AS_PATH (más a la derecha). Un ejemplo de este proceso se encuentra en la sección siguiente.

Capítulo 3

Técnicas basadas en BGP

Así como en la capítulo anterior se describieron técnicas que utilizaban mayoritariamente traceroutes para construir topologías, en este se explicara el uso de BGP para obtener topologías a nivel de sistemas autónomos.

3.1. Generalidades

BGP (Border Gateway Protocol) [56], es el protocolo por defecto en Internet usado para intercambiar información sobre conectividad y definir rutas entre Sistemas Autónomos.

El protocolo a su vez se divide en dos: eBGP (external BGP) e iBGP (interal BGP). La función del primero es intercambiar información de ruteo con otros ASes, mientras que la del segundo es distribuir la información aprendida usando eBGP dentro del AS.

Denominaremos BGP speaker a aquellos routers que implementen BGP en cualquiera de sus dos versiones.

Dos BGP speakers en distintos ASes (external peers), establecen una conexión BGP a través de la cual se realizará el intercambio de información. Esta consiste principalmente en prefijos, a los que uno de los peers sabe como alcanzar, complementado con información sobre la ruta hacia el mismo.

Estos routers implementan tres tipos de bases de datos en las que almacenan la información: Adj-RIB-In, Adj-RIB-Out y Local-RIB. Teóricamente, hay una base Adj-RIB-IN y una Adj-RIB-Out para cada peer, pero esto no es obligatorio en la práctica.

La primera consiste en toda la información recibida por los external peers,

previo al procesamiento. La segunda consiste en aquellas rutas que van a ser publicadas a otros ASes (filtrada por la política que implementada en el AS) y la última consiste en las rutas que fueron elegidas por el proceso de decisión.

Cuando dos external peers establecen su conexión por primera vez, se envían el contenido de su tabla Adj-RIB-Out. Todos los futuros cambios en las rutas son enviados mediante mensajes UPDATE (esto incluye nuevas rutas o la eliminación de viejas rutas).

3.2. Formato de los mensajes

Existen varios tipos de mensajes definidos en BGP: Open, Update, Notification y Keep Alive. Dedicaremos principal atención a los del tipo UPDATE.

Los mensajes de tipo UPDATE tienen dos funciones en BGP: advierten de nuevas rutas, pero también de rutas que deben ser removidas (WITHDRAW).

Las rutas que son removidas son informadas simplemente utilizando el prefijo de las mismas. Por otra parte, cada nuevo prefijo es acompañado de algunos atributos: los más importantes siendo: NEXT_HOP, AS_PATH y ORIGIN.

El atributo NEXT_HOP indica la IP por la cual se accede a la ruta en cuestión.

El atributo ORIGIN explicita donde se originó la información (dentro del AS, fuera del AS, o por algún otro medio).

El atributo AS_PATH es una lista de identificadores de ASes, que indica el camino (a nivel de ASes) que se debe realizar para llegar al destino. Cuanto más reciente es un identificador, más hacia la izquierda se encuentra. Cada vez que un AS envía una ruta, añade su número de AS al principio del AS_PATH.

3.3. RouteViews y RIPE

RouteViews es un proyecto de la Universidad de Oregon [45] para recolectar información sobre el funcionamiento de BGP.

Con este propósito, conectan routers mediante sesiones BGP con otros sistemas autónomos. A pesar de no enviar prefijos propios, reciben los men-

TIME: 04/01/17 00:00:47 TYPE: BGP4MP/MESSAGE/Update FROM: 213.144.128.203 AS13030 TO: 128.223.51.102 AS6447 WITHDRAW 2.94.27.0/24 137.128.0.0/16

TIME: 04/01/17 00:15:00 TYPE: BGP4MP/MESSAGE/Update FROM: 103.247.3.45 AS58511 TO: 128.223.51.102 AS6447 ORIGIN: IGP ASPATH: 58511 6939 198371 206976 NEXT_HOP: 103.247.3.45 ANNOUNCE 185.169.231.0/24

Cuadro 3.1: Mensajes UPDATE en el formato MTR obtenido de [45]. A la izquierda se puede observar un withdraw mientras que a la derecha se advierte una nueva ruta.

sajes UPDATES enviados por sus peers: estos mensajes son públicos, por lo que se pueden descargar.

Por otra parte, cada cierto tiempo realizan una descarga de su tabla Local-RIB, ésta también se encuentra disponible en su página web.

RIPE [43] actúa como el registro regional de Internet (RIR) para Europa, Medio Oriente y Asia Central. Entre sus muchos proyectos se encuentra RIPE RIS [44], el cual recolecta información sobre el routeo BGP en una manera similar a RouteViews.

3.4. Looking Glasses

Otra alternativa para conseguir entradas de una tabla BGP es por medio de Looking Glasses.

Un Looking Glass es un router que ofrece una interfaz web para ejecutar algunos comandos. Los más típicos traceroute, aunque los hay también para: *show bgp ip* y *show bgp summary*.

3.5. Infiriendo mapeo entre IP y ASN

Como se mencionó brevemente al final del capítulo 2, es posible usar los mensajes BGP para inferir a que sistema autónomo pertenece una IP. Para

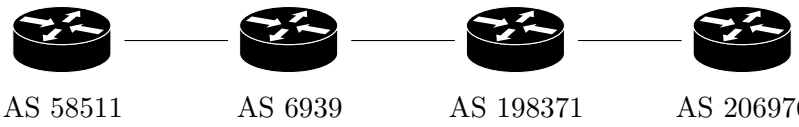


Figura 3.1: Topología inducida a partir del Cuadro 3.2

esto se supone que todas las IP dentro de un prefijo X/Y pertenecen al primer AS en el AS_HOP, es decir, el que se encuentra más a la derecha. Usando el mensaje UPDATE ubicado en el Cuadro 3.2 como ejemplo, podríamos inferir que el todas las IPs dentro del prefijo 185.169.213.0/24 pertenecen al AS con número 206976.

3.6. Infiriendo una topología

Las trazas BGP son particularmente útiles para inferir una topología a nivel de sistemas autónomos. Para esto, basta tan solo con observar el AS_PATH. Si dos identificadores aparecen consecutivos en el mismo, entonces existe un link entre ellos. Por ejemplo, del UPDATE a la derecha en el Cuadro 3.2 podríamos inferir la topología de la Figura 3.1.

3.7. Relaciones entre ASes

El tráfico entre dos ASes depende del tipo de intercambio comercial entre los ISPs a los que pertenecen. En este sentido, Gao [16] propuso la siguiente caracterización:

- Proveedor-Consumidor: En este tipo de relación, el ISP consumidor debe pagar por tránsito al ISP proveedor.
- Peer-Peer: Ambos ISPs acceden a intercambiar trafico sin cobrarse mutuamente.
- Hermano-Hermano: Son utilizados para conectar dos ASes que pertenecen a un mismo ISP.

y propuso un algoritmo para inferir las mismas. Algunos trabajos posteriores formalizaron el problema y se probó que es NP difícil [12].

Dado que se trata de información sensible para los ISPs, generalmente no se encuentra disponible de forma pública y se debe medir.

Utilizando algunos de estos algoritmos, CAIDA (Center for Applied Internet Data Analysis) generó una base de datos con relaciones entre ISPs que puede ser descargada de su página web [53].

3.8. Problemas al usar BGP

A pesar de presentar un marco natural para inferir topologías a nivel de sistemas autónomos, este procedimiento presenta algunas dificultades.

Uno de los primeros problemas es que BGP está diseñado para filtrar información basado en políticas. En este sentido, cada ISP decide que rutas transmitir y cuales no. Esto ocasiona que muchas rutas existentes (a nivel físico) nunca sean anunciadas y por lo tanto permanezcan invisibles. Por ejemplo: si dos ASes establecen una relación Peer-Peer, está será anunciada entre ellos y no hacía otros ASes (de ocurrir esto, un tercer AS podría usar esa ruta como tránsito y no es el propósito).
Dado que los servidores BGP disponibles para realizar mediciones son mucho más escasos, es muy probable que ninguno de los dos sean un punto de medición y en consecuencia, la ruta quedará invisible.

Otro problema se presenta cuando un AS agrega identificadores al AS_PATH. Esta práctica no está prohibida y puede ocasionar la inferencia de caminos que no son reales.

Por último, y enfatizando el primer problema, un estudio realizado por B.Ager et.al [2] en un IXP europeo mostró que las topologías a nivel de ASes construidas hasta el momento presentaban un número bajo de aristas p2p.

Capítulo 4

Modelos

El interés de la comunidad científica en obtener buenos modelos matemáticos de la topología de Internet es bastante directo: buenos modelos permiten probar nuevas hipótesis sin tener que realizar mediciones sobre la red, lo que como se vio en secciones anteriores, no es sencillo.

A nivel de topología de Internet, los modelos toman la forma de grafos, un marco natural para representar una red. En el modelo no se busca representar de forma exacta todos los actores involucrados, sino que se busca abstraer las propiedades más importantes para que la red modelada sea lo más *verídica* posible.

Existen dos grandes familias de modelos: los estáticos y los dinámicos. Los modelos estáticos mantienen fija la cantidad de vértices, mientras que los dinámicos definen procesos mediante los cuales está cantidad puede cambiar.

Está claro que dada la naturaleza cambiante de Internet (routers que fallan, sistemas autónomos que se crean, etc) los modelos dinámicos son más fieles a la realidad. De todas formas, los modelos estáticos ofrecen un marco bastante poderoso para generar modelos.

A lo largo de este capítulo se seguirá como referencia el libro de Pastor-Satorras y Vespignani [41].

4.1. Propiedades deseables

No todos los grafos aleatorios con N vértices son buenos modelos de Internet. Para poder decidir si un modelo es más apto que otro, debemos medir ciertas propiedades tanto del modelo como de Internet y encontrar

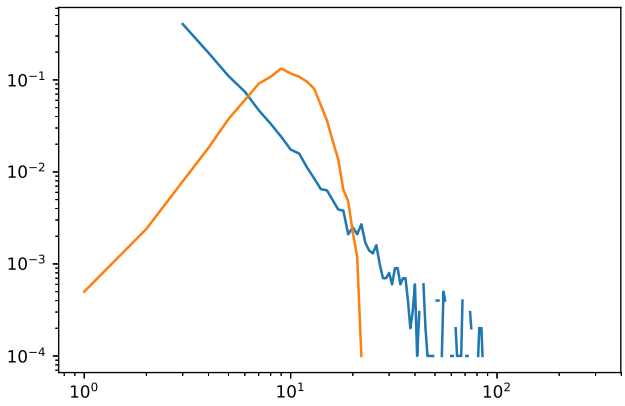


Figura 4.1: Dos distribuciones de grado en escala log-log. La azul se aproxima por una recta (a pesar del ruido sobre el final) mientras que la naranja presenta una caida exponencial.

aquellos modelos que mantengan las propiedades más importantes.

Que propiedades presenta Internet estudiado como un grafo (a nivel de routers o ASes) es un tema de mucho debate en el área.

4.1.1. Métricas

Sea $G = (V, E)$ un grafo no dirigido con $n = |V|$ nodos y $m = |E|$ aristas. Usando una notación proveniente de la Física, denotaremos $\langle X \rangle$ al promedio de la propiedad X.

Distribución de grado

Para cada grado k se calcula la probabilidad de que un nodo tenga grado k y se denota $p(k)$. Lo relevante es estudiar que tipo de función es $p(k)$. Para esto se grafica $p(k)$ en función de k en escala logarítmica (Figura 4.1). En el caso de las redes complejas y en particular Internet, se ha observado que en escala log-log, la distribución aproxima una recta con coeficiente $-\gamma$. Cuando este fenómeno ocurre se dice que $p(k)$ sigue una power-law o más formalmente $p(k) \sim k^{-\gamma}$. Para el grafo de ASes, el coeficiente γ se ha mantenido constante a lo largo de los años con un valor aproximado de 2,1.

Camino más corto

El camino más corto entre dos vértices v y w es la cantidad de aristas en el camino más corto entre v y w y se denota l_{vw}. Se a observado que $\langle l \rangle \sim \log(n)$ o incluso $\langle l \rangle \sim \log(\log(n))$. Este efecto se conoce como *mundo pequeño*, característico de las redes complejas y famoso por el trabajo: *6 grados de separación*. Hallar $\langle l \rangle$ implica encontrar todos los caminos más cortos y esto tiene una complejidad $O(n^3)$.

Coeficiente de clustering

El coeficiente de clustering c_v se define como la cantidad de vecinos de un nodo que son a su vez vecinos entre ellos. Si $\langle m_{nn}(k) \rangle$ cuenta el promedio de vecnios de nodos de grado k que son vecnios entre ellos, entonces el coeficiente de clustering local de un nodo de grado k es: $c(k) = \frac{\langle m_{nn}(k) \rangle}{k(k-1)}$. El coeficiente de clustering global se define como: $C = \sum_i c(i)p(k)$. La complejidad de calcular el coeficiente de clustering global es $O(n^3)$.

Distribución de grado conjunta

La probabilidad de grado conjunta se define como la probabilidad de que exista una arista entre un nodo de grado k_1 y otro de grado k_2. Está queda totalmente caracterizada por la matriz de correlación de grado. Está métrica contiene más información que la distribución de grado pura, pues aparte agrega información sobre la conectividad. Una estadística que resume la anterior es el grado promedio de los vecinos de un nodo de grado k ($k_{nn}(k)$). En [30] proponen que la distribución de grado conjunta es la mejor métrica para estudiar la topología de Internet pues de ella se pueden deducir la distribución de grado, el grado promedio así como estimar la magnitud de otras propiedades.

4.1.2. Descomposición en k-cores

Le llamaremos $k - core$ al máximo subgrafo inducido H en G tal que todos sus vértices tienen grado k o más. Un vértice v_i tiene *coreness* c si pertenece al $c - core$ pero no al $c + 1 - core$. Está descomposición es fácil de calcular (se remueven en orden los vértices de grado más chico) y permite visualizar grandes rasgos. En [3], se discute como simplemente observando

Grafo	Método	$\#V$	$\#E$	$\langle k \rangle$	máx(k)	C	γ_1	γ_2
Ark ITDK ASro	traceroute	25,578	66,401	5.19	2,607	0.33	2.19	2.18
RouteViews2 AS	BGP	37,606	80,051	4.26	3,100	0.21	2.15	2.12
BGP full AS	BGP	36,876	103,481	5.61	2,972	0.24	2.12	1.97

Cuadro 4.1: Algunos grafos a nivel de AS utilizados en el estudio [21]. γ_1 es el exponente de la ley de potencias estimado usando mínimos cuadrados mientras que γ_2 se calcula usando máxima verosimilitud.

la representación se pueden extraer características cualitativas de la red (y como se diferencia una topología a nivel de routers o de AS, por ejemplo).

4.1.3. Valores de métricas en Internet

El artículo seminal de los hermanos Faloutsos [14] en 1999 encontró que la distribución de grado sigue una ley de potencias, es decir que la probabilidad $p(k)$ de que un nodo tenga grado k cumple que $p(k) \propto d^{-\gamma}$ con $\gamma = 2{,}1$. Desde entonces muchos trabajos han encontrado evidencias de una distribución bajo ley de potencias en Internet. Hasta el momento, el modelo más utilizado era el Erdös-Réni. Sin embargo, la distribución de grado de dicho modelo no se adecua a una ley de potencias y por lo tanto fue necesario buscar nuevos modelos.

Aunque existen muchos trabajos que evidencian las propiedades anteriores, existe preocupación en la literatura de que las propiedades antes mencionadas sean artífices de sesgos en las mediciones. Lankhina et.al. [26] mostraron que mediciones similares a traceroutes sobre un grafo aleatorio producen una distribución que se asemeja a una ley de potencias, poniendo en tela de juicio los resultados de Faloutsos. En 2006, Hamelin et.al. [9] propusieron que el fenómeno descrito por Lakhina es poco probable en una red como Internet.

A pesar de que se trata de una área con controversia, se ha ido avanzando en modelos dinámicos que crezcan de forma similar a la esperada en Internet y que presenten las propiedades antes mencionadas.

En el cuadro 4.1 se pueden observar los valores observados de distintas métricas para algunas mediciones de Internet. Usaremos el trabajo original [21] para evaluar los distintos modelos a medida que son presentados.

4.2. Primeros Modelos

4.2.1. Erdös-Rényi

El modelo Erdös-Rényi fue el primer modelo de grafo aleatorio y se considera que con él nace lo que se conoce hoy día como: Network Science.

Aunque la formulación original era levemente distinta, denotaremos un grafo Erdös-Rényi como $G_{N,p}$ El grafo $G_{N,p}$ es un grafo con N vértices donde la probabilidad de cada arista de estar en el grafo o no son variables $X \sim Be(p)$ i.i.d.

En particular para $G_{N,p}$ se tiene que $\langle k \rangle = p(N-1) \approx pN$. Como el grado esperado tiende a infinito y esto en general no sucede en redes reales, en general se toma la probabilidad p en función de N como $p(N) = \frac{\langle k \rangle}{N}$. Bajo estas hipótesis, la distribución de grado $P(k)$ sigue una Poisson cuando N tiende a infnito

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

$$N \to \infty \quad P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$$

Por último se prueba que el coeficiente de clustering es $\langle c \rangle = \frac{\langle k \rangle}{N}$.

Podemos ver que el coeficiente de clustering esperado para cualquiera de las topologías del Cuadro 4.1 sería del orden de: $\frac{5}{10000} = 0{,}0005$ y esto es casi 3 ordenes de magnitud por debajo de lo observado.

Bajo esta evidencia y el hecho de que una distribución Poisson no es similar a una ley de potencias, desestimamos el modelo.

Los detalles de las derivaciones se encuentren en el Apéndice A.

4.2.2. Configuración

El modelo de configuración puede verse como una modificación del Erdös-Renyi en la que la secuencia de grado se puede elegir de forma arbitraria (como una ley de potencias por ejemplo), pero por lo demás se comporta como un grafo aleatorio.

En particular, no existe correlación entre los vértices y se puede probar [36] que el coeficiente de clustering difiere tan solo en una constante con el modelo anterior, por lo que tampoco es apto para modelar Internet.

4.3. Generadores de Internet

A continuación presentaremos algunos modelos utilizados por la comunidad informática para generar topologías de Internet en las cuales probar algoritmos.

4.3.1. Waxman

El primero de estos algoritmo y utilizando durante mucho tiempo fue propuesto por Waxman [55] basado en dos argumentos:

- Los routers están distribuidos en el espacio algunos cerca y otros lejos.
- La conexión entre routers lejanos es menos probable que entre routers cercanos.

En base a estos principios, el algoritmo propuesto consiste en:

1. Sortear N vértices en un cuadrado de lado L de forma aleatoria
2. Sortear cada arista con una probabilidad $p_{ij} = \beta \exp(-\frac{d_{ij}}{\alpha L})$, donde β controla la cantidad de aristas, d_{ij} es la distancia euclidea entre ambos vértices y α permite ajustar que tan probable es una distancia dada.

El modelo presenta dos problemas: no garantiza la conectividad del grafo, por lo que es posible tener que simular varias veces hasta obtener un grafo conexo y tiene una distribución de grado que no sigue una ley de potencias.

4.3.2. Brite

Brite (Boston Univeristy Representative Topology Generator) [32] es un conjunto de generadores de topologías de Internet que está en uso al día de hoy.

Las variantes de los distintos modelos se pueden resumir en tres etapas:

1. Colocar los nodos en el plano.
2. Asociar las aristas.
3. Asignar ancho de banda a las aristas.

Para el paso (1), hay dos opciones: se agregan de forma aleatoria en el espacio o se crea una grilla y en cada grilla se agregan tantos nodos como el resultado de sortear una variable aleatoria que siga una ley de potencias (un valor distinto para cada cuadrado en la grilla).

Para el paso (2) puede utilizar una probabilidad similar a la propuesta por Waxman (discutido arriba) o Barabási-Albert (presentado en la siguiente sección).

Para el último paso, la asignación puede ser: a) constante, b)uniforme, c)exponencial o d) de cola pesada.

Brite permite generar tanto topologías a nivel de routers o de ASes. No hay grandes diferencias más que a nivel de sepraración en el código entre los dos tipos. Por último, permite introducir jerarquía en las simulaciones. Para lograr esto simplemente itera sobre el proceso y *las une*.

4.4. Topologías dinámicas

4.4.1. Barabási-Albert

Un modelo que estableció un antes y después fue el modelo de Conexión Preferencial [7] propuesto por Barábasi y Albert en 1999.

El cambio más radical se encuentra en que las nuevas aristas no se agregan de forma aleatoria sino que tienen más probabilidad de unirse a vértices con mayor grado (efecto *richers get richer*).

El modelo se puede resumir en las siguientes reglas:

- El grafo comienza con m_0 vértices y a cada instante se agrega un nuevo vértice con m aristas $m < m_0$ las cuales son conectadas a los viejos vértices en el sistema.
- La probabilidad de que una arista sea conectada a un vértice s es proporcional a su grado k_s.

El modelo puede ser representado en el marco de redes dinámicas por la ecuación diferencial (done la notación $k_s(t)$ significa el grado promedio del nodo s en el tiempo t):

$$\frac{\delta k_s(t)}{\delta t} = \frac{m k_s(t)}{2mt + 2m_0\langle k \rangle_0} \tag{4.1}$$

esto permite calcular la distribución de grado en el límite ($t \to \infty$):

$$P(k) = 2m^2 k^{-3}$$

donde se observa una ley de potencias con exponente constante $\gamma = 3$. Lamentablemente, el coeficiente de clustering:

$$\langle c \rangle = \frac{m}{8N}(lnN)^2$$

tiende a 0 como N^{-1} (la corrección logarítmica es despreciable para N grande).

Existen modificaciones del modelo que permiten modificar el parámetro γ de la ley de potencias. La dedución de la distribución de grado se encuentra en el Apéndice A.

4.4.2. HOT :)

Un óptica diferente para la evolución de los modelos es considerar las conexiones de los nuevos vértices como un problema de optimización.

En cada instante se añade un vértice nuevo en una posición aleatoria del cuadrado unidad y se agrega una arista entre el nuevo vértice i y el vértice j que minimice la ecuación:

$$\Psi(i,j) = \alpha(N)d(i,j) + \phi(j)$$

donde $\alpha(N)$ es una constante que depende del tamaño final de la red, $d(i,j)$ es la distancia euclidea y $\phi(j)$ es una medida de centralidad del vértice j.

Dependiendo del parámetro $\alpha(N)$, la topología resultante puede presentar una distribución que sigue una ley de potencias. A pesar d esto, es fácil ver que el modelo genera árboles y por lo tanto el coeficiente de clustering es nulo.

Capítulo 5

Proyectos Destacados

En este capítulo se describirán algunos proyectos o campañas particularmente relevantes a la construcción de topologías de Internet. Se describe mayormente los esfuerzos de CAIDA por construir topologías así como otros proyectos independientes dignos de mencionar.

5.1. Campañas de medición

A lo largo del tiempo se han realizado diversos esfuerzos en medir la topología de Internet. Dado que las ejecuciones de traceroute distribuido requieren de puntos de medición lo más separados posibles, no es trivial instalar la infraestructura necesaria.

5.1.1. CAIDA

CAIDA (Center for Applied Internet Data Analysis), fue uno de los pioneros en establecer un sistema distribuido de medición (Skitter) [49] el cual recolectó información entre 1998 y 2008, antes de transicionar al actualmente en funcionamiento: Archipielago (Ark) [4].

Scamper

Scamper es un software desarrollado por CAIDA [29] para realizar mediciones a nivel global. Una de sus ventajas es que integra implementaciones de muchas herramientas utilizadas, estandarizando los formatos entre

ellas. A modo de ejemplo: permite realizar una campaña distribuida de Paris-Traceroute y luego resolver aliases utilizando una variante del método analítico o el método basado de dirección. Actualmente es el software utilizado en su proyecto Archipielago.

5.1.2. RIPE

Actualmente cuenta con casi diez mil sondas distribuidas en todos los continentes. Más información puede encontrase en su página web [43].

5.2. Otros proyectos más pequeños

Como ha sido mencionado, en el área existe controversia sobre la corrección de las mediciones realizadas, en particular en cuanto al sesgo en las medidas y a las posibles falsas conclusiones a las que se llegan.

Muchos trabajos basan sus resultados en datasets poco documentados que, dada la realidad compleja que se desea modelar, podrían incurrir en defectos estructurales que invaliden todos los análisis posteriores.

Por ejemplo: tomar un dump Local-Rib de BGP creado por RouteViews y asumir que está completo porque en el están representados casi todos los ASes sería una falla de las que hablamos. Hay, por supuesto, otras más sutiles como la ubicación de los puntos de medición, el software utilizado, etc.

En la más reciente revisión del tema [35], Motamedi et.al. destacan algunos trabajos por su rigurosidad y buenas prácticas. A continuación se presentará un breve resumen de dichos proyectos.

5.2.1. IXP: Mapped?

Augustin et.al [5] observaron que los IXP (Internet Exchange Point) son un componente clave en la topología a nivel de sistemas autónomos, pero que por particularidades técnicas era muy probable que no se estuviesen registrando en estudios de talla global como el proyecto Ark.

En un IXP, varios ASes colocan routers interconectados a nivel de la capa 2 (por un switch, por ejemplo) y se le asigna a cada interfaz conectada de esta forma una IP única característica del IXP.

Para lograr su objetivo (descubrir enlaces entre ASes en IXPs) primero recolectaron información sobre que IXPs existen y quienes son sus miembros

(está información es generalmente pública). Luego, obtuvieron una lista de LG y seleccionaron los que se encontraban en los ASes que querían medir. Para cada AS en cada IXP, utilizaban un LG que estuviese en ese AS de ser posible o en un AS a lo sumo 2 saltos de distancia (en una topología a nivel de ASes confeccionada usando RouteView y RIPE RIS). Notesé la complejidad de modificar el punto de medición para cada (AS,IXP) con tal de tener mayor seguridad en las mediciones.

Por otra parte, conscientes de las imperfecciones a la hora de resolver direcciones IP en ASes, establecieron varios niveles de confianza en sus datos para finalmente descartar todos aquellos datos que no estaban en el nivel más alto.

El resultado de dicho trabajo también es sorprendente, encontraron 44.000 conexiones (específicamente en IXPs) de las cuales 18.000 son nuevas. Si consideramos que un proyecto como BGP full (Cuadro 4.1) tiene 100.000 aristas, este estudio encontró un 10 % de aristas faltantes así como algunas más que saben de su existencia pero no pudieron descubrir.

5.2.2. Dasu

Uno de los mayores problemas que presentan las plataformas de medición más populares (Ark, PlantLab, etc) es que la distribución de sus nodos (demasiado *céntrica*) no permite observar lo que ocurre en la periferia de la red (más cerca del usuario final). Otros proyectos, como por ejemplo RIPE Atlas, permiten que usuarios finales administren puntos de medición pero los autores de [46] argumentan que existe un conflicto de intereses entre los usuarios que hostean los puntos de medición y aquellos que desean hacer uso de los mismos.

En este contexto y como solución proponen la plataforma Dasu, una extensión de BitTorrent (!) que implementa una API para poder lanzar desde simples traceroutes hasta consultas mucho más complejas. A cambio de descargarse la extensión, ofrecen información sobre la calidad del ISP que provee internet a cada usuario (que es actualizada utilizando la herramienta) y esto a sido suficiente para que casi 90.000 usuarios en todos los continentes hagan uso de la misma.

Algunas de las ventajas de su solución es que al ser una extensión de Bit Torrent, existe una gran disponibilidad de los puntos de medición.

Utilizando esta plataforma, pueden obtener varios puntos de medición dentro de un mismo AS y usan este hecho para probar que los resultados

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	prefixes
x	x	-	-	-	-	-	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	x	x	A
✓	✓	✓	✓	✓	-	-	-	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	B

Table 4: *Prefix-based peering at Amsterdam Internet Exchange (AMS-IX) between two ASes. Columns show the hour, local time. Legend: '✓' probes crossed IXP; 'x' probes did not cross IXP; '-' no probes.*

Figura 5.1: Evolución y visibilidad de dos prefijos a lo largo del día.Figura extraída de [46]

encontrados en IXP: Mapped? (basados en una única medición de traceroute) no son exactos y que la realidad es mucho más dinámica. Efectivamente en la Figura 5.1 se puede observar la diferencia entre dos prefijos (A y B) en el mismo AS.

Capítulo 6

Exploración en NS3

Buscando familiarizarse con las metodologías descritas en los capítulos anteriores, se optó por desarrollar un ambiente de simulación desde el cual es pudiesen realizar campañas de medición arbitrarias.

El objetivo de estas exploraciones no fue trabajar sobre la red más real posible, sino lograr operar a tan bajo nivel como fuese necesario para lograr comprender todos los elementos involucrados en las mediciones. Es por esta razón que la realidad simulada presenta diversas simplificaciones respecto a una topología real, lo cual, nuevamente, no va en desmedro del objetivo de la simulación.

El capítulo se divide en tres pares: en la primera se explicarán las características tecnológicas del entorno desarrollado, en la segunda las características de la topología simulada así como las técnicas utilizadas y por último se discuten algunos resultados encontrados.

El código del proyecto así como una documentación básica para su uso puede encontrarse en [1].

6.1. Tecnologías Utilizadas

A la hora de realizar experimentos sobre redes, existen tres caminos posibles: trabajar sobre la red real, trabajar sobre una red emulada o una simulada.

Las ventajas y desventajas de trabajar sobre una red real son bastante evidentes, aunque es importante resaltar que en el caso de Internet, no existe

[1] `https://github.com/ellguso/topology_simulation`

ground truth para validar experimentos de gran porte.

6.1.1. NS3 y DCE

La decisión entre redes emuladas (ej: Netkit [42]) y simuladas (ej: NS3 [37]) se resume en la mayoría de los casos a corrección versus escalabilidad.

Dado que Internet cuenta con una cantidad muy grande de nodos y que muchas de las dinámicas se pierden al trabajar con unos pocos, se optó por la vía de la simulación, específicamente NS3.

NS3 es un simulador de eventos discretos desarrollado específicamente para experimentar sobre Internet. A grandes rasgos, NS3 cuenta con un planificador que agenda y ejecuta eventos, mayoritariamente relacionados con la creación y recepción de paquetes.

Uno de los principales contratiempos encontrados a la hora de implementar simulaciones en NS3 fue la falta de herramientas similares a Traceroute. Una de las alternativas era implementar Traceroute utilizando las primitivas propias del entorno aprovechando que es software libre, sin embargo esto alternativa resulta poco eficiente.

La alternativa por la que se optó fue utilizar DCE (Direct Code Execution). DCE consiste en una extensión de NS3 que permite ejecutar programas externos al entorno de simulación (como Traceroute). DCE es muy cómodo de utilizar pues en lugar de tener que reescribir código fuente, *camufla* las llamadas a funciones del sistema (POSIX) por llamadas a funciones propias de NS3, no teniendo que escribirse nuevo código. Por ejemplo, cuando un programa llama a la función del sistema **gettimeofday**, la llamada se redirige a una propia de NS3 que devuelve el tiempo en la simulación y no el real del SO. Una descripción más detallada puede encontrarse en [25].

6.1.2. Docker y Paris-Traceroute

Dado que uno de los factores más importantes en una simulación es su reproducibilidad, se decidió que el entorno de desarrollo estuviese dockerizado [33]. Idealmente se hubiese utilizado el programa Scamper sobre DCE ya que permite realizar campañas de medición y resolución de aliases, sin embargo, el proceso de configuración (dentro del ambiente dockerizado) se tornó muy complejo y por restricciones de tiempo se decidió utilizar solamente Paris-Traceroute. Está decisión tiene como clara desventaja que la complejidad de los experimentos realizables disminuye mucho, pero por otra parte, brindó la

oportunidad de desarrollar algunos algoritmos que hicieron a la comprensión des herramientas y metodologías.

6.2. Realidad Modelada y Procesamiento de los Datos

A la hora de programar el entorno de simulación, se tomaron decisiones de diseño que son importantes de tener en cuenta a la hora de analizar los resultados obtenidos. Las topologías implementadas consisten de routers interconectados por enlaces punto a punto. Esto es una simplificación de la realidad en donde varios routers pueden encontrase conectados en un mismo canal CSMA.

Para la asignación de direcciones IP (la cual se debe hacer manual), se optó por asignar a cada enlace punto a punto una subred con prefijo 30, pero sin ningún tipo de jerarquía basada en la topología modelada.

La elección de la cantidad de sondas así como su ubicación es un parámetro totalmente configurable. Por otra parte, se optó por utilizar todas las IPs conocidas como destinos. Para entender el comportamiento cuando no todas las direcciones son destinos, el algoritmo de resolución de aliases permite filtrar de forma aleatoria cierta cantidad de trazas. Es decir, se simula todo y luego se muestrea para analizar.

Una vez obtenidos el resultado de la simulación (salida estándar de Paris-Traceroute), se procedió a implementar un algoritmo de resolución de aliases. La versión implementada está basada en el algoritmo APAR [19].

En la implementación del algoritmo se intentó evitar aquellos pasos que permiten la resolución de aliases errados, prefiriendo calidad a cantidad. El primer paso para esto fue ignorar todos los caminos que presentaban routers anónimos o loops.

En una segunda instancia, se eliminó el último salto en cada traza. Esto es necesario ya que dada la implementación usada, el destino del Traceroute responde con la dirección final y no necesariamente con la de la interfaz más cerca, provocando falsas inferencias.

Una vez que las trazas se encontraban sanitizadas, se procedía a encontrar direcciones IP en una misma subred, estas son usadas para alinear las trazas y resolver alias. Dado que todos los enlaces son punto a punto, la verificación de *vecino común* (condición número cinco en [19]) no es necesaria. Si que lo

es, por otra parte, que la asignación de aliases no genere un loop en cualquiera de las dos trazas.

A pesar de que el algoritmo se adaptó de forma tal que no permitiese la resolución de aliases incorrectos, experimentando se encontró que hay routers que no siempre responden con la dirección de la interfaz esperada, provocando errores en la resolución. Este problema no puede ser solucionado.

6.3. Resultados experimentales

Con motivo de aplicar el entorno de simulación desarrollado, se construyeron 100 topologías utilizando el generador *barabasi_albert* de la biblioteca de Python Networkx [10] con parámetros $N = 50$ y $m = 3$. La elección está motivada principalmente porque es uno de los modelos más simples que presentan ley de potencias y se deseaba estudiar la aparición de las mismas. Los grafos en cuestión cuentan con 50 vértices (duh!) y 141 aristas. Se realizaron 100 simulaciones, variando la cantidad de sondas en cada una (dos simulaciones para cada cantidad de sondas).

Las 100 simulaciones tomaron aproximadamente 50 horas.

Para aumentar la riqueza de los datos, para cada simulación se consideró el resultado con todas las interfaces como objetivo (resultado estándar de la simulación), el mismo con tan solo la mitad de los objetivos elegidos al azar y con el 10 %, también elegido al azar. Finalmente el análisis a continuación está compuesto por 300 resultados con una cantidad variable de sondas así como de objetivos en una topología con cantidad fija de vértices y aristas.

6.3.1. Completitud

Uno de los estadísticos más importantes a la hora de realizar mediciones es lograr descubrir la mayor parte (toda) de la topología subyacente. En nuestro caso, al tener la topología real, es posible hallar el porcentaje de cubrimiento obtenido. Para esto, basta observar que la máxima cantidad de nodos observables en el grafo previo a la resolución de aliases es igual a la cantidad de interfaces: 282, dos veces la cantidad de aristas.

Como era esperable, en la Figura 6.1 se puede observar que al usar todas las interfaces como objetivos, todas son descubiertas con cualquier cantidad de sondas. El resultado que es más interesante se da al utilizar el 10 % de los objetivos totales. En este caso, con el 10 % de la topología como sondas

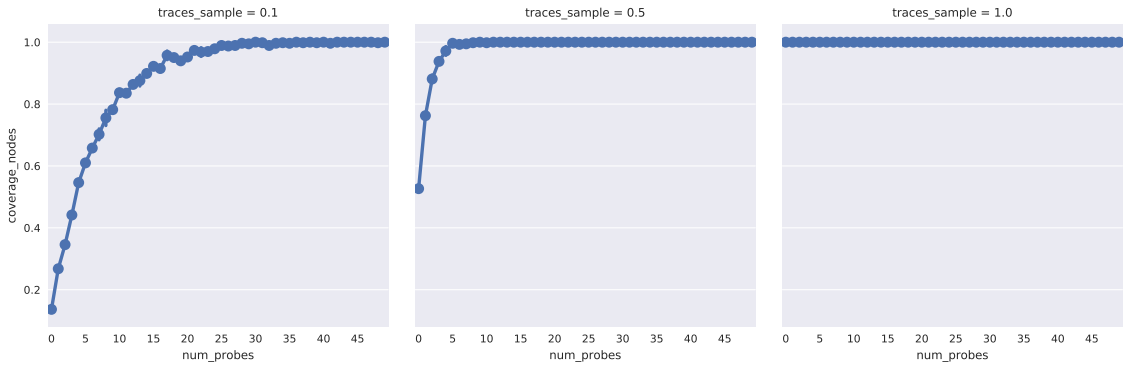


Figura 6.1: Cantidad de interfaces encontradas a medida que aumenta la cantidad de sondas, desglosada por la cantidad de destinos

(número mucho mayor que en estudios reales), no se logra descubrir ni la mitad de las interfaces totales. Lo que es más, son necesarias casi 25 sondas (50 % del total) para alcanzar un cubrimiento cercano al 90 %.

Dado el grafo no resuelto, es posible hallar (usando la topología original), cuántas de sus aristas se corresponden con aristas reales entre nodos en el grafo inicial. De esta forma obtenemos una métrica más precisa de la cantidad de aristas encontradas que simplemente contar el número hallado en el grafo no resuelto.

La Figura 6.2 permite encontrar un resultado interesante: aumentando la cantidad de objetivos y manteniendo fija la cantidad puntos de medición, se puede encontrar hasta un 20 % más de aristas. Esto no es menor ya que la cantidad de objetivos es de los pocos parámetros *fáciles* (recordar esta afirmación en el capítulo 8) de modificar en una campaña de medición.

Sin embargo, al menos en las simulaciones realizadas, aumentar la cantidad de objetivos compromete la calidad de las inferencias. Como habíamos mencionado, muchas veces un router no responde con la interfaz esperada y esto provoca falsos positivos. En la Figura 6.3 se puede ver como la precisión desciende tanto con la cantidad de sondas como con la cantidad de objetivos. Esto es esperable en la medida de que cuantas más interacciones se produzcan, más probable es encontrarse con comportamientos no deseados.

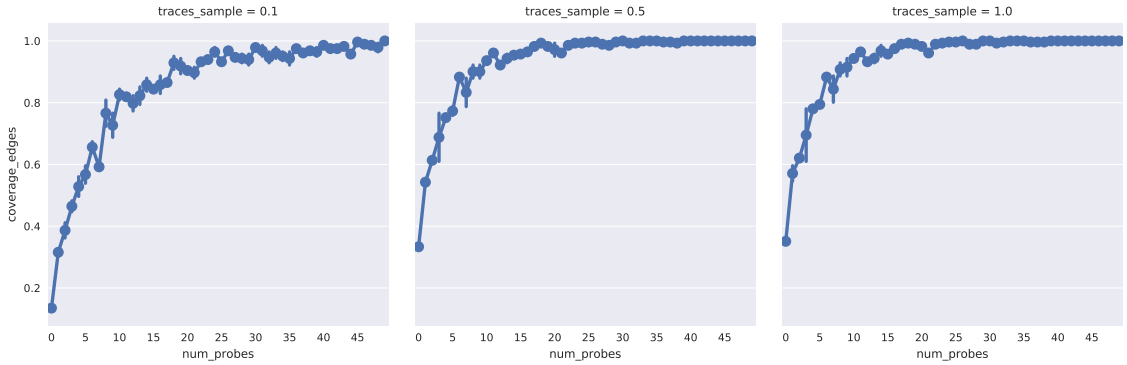


Figura 6.2: Cantidad de aristas en el grafo no resuelto que se corresponden con aristas de la topología real

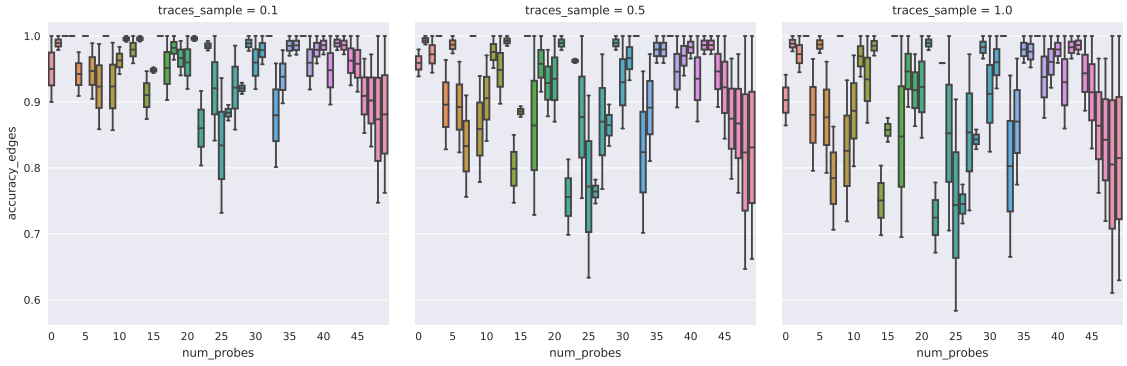


Figura 6.3: Porcentaje de las aristas descubiertas que se corresponden con aristas reales en la topología original

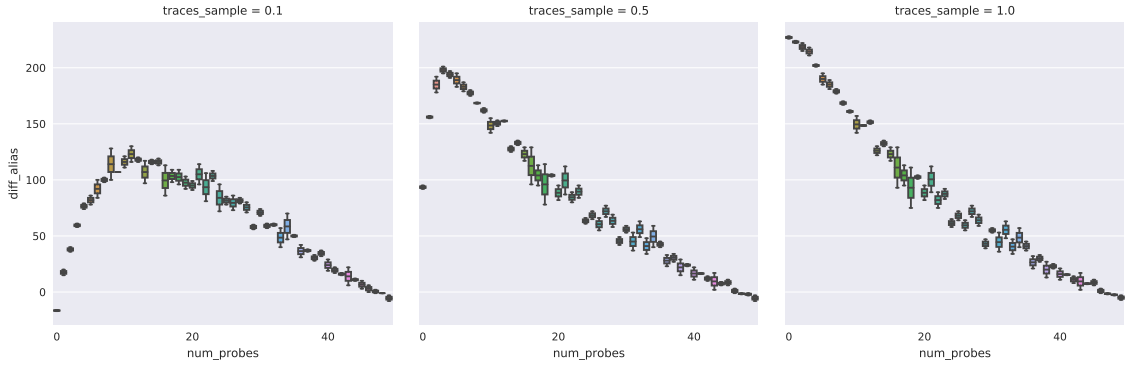


Figura 6.4: Nodos adicionales a los que no se les encontró alias en función de la cantidad de sondas utilizadas

6.3.2. Capacidad de Resolución

Dado que las mediciones con Traceroute no obtienen routers sino interfaces y hay que utilizar heurísticas para obtener los primeros a partir de los segundos, es vital comprender que tan *resuelto* se encuentra el grafo final. Observar que una completitud mayor no necesariamente implica un buen poder de resolución. Esto puede ser particularmente perjudicial porque compromete nuestra capacidad de inferir propiedades sobre la topología. La estrategia más simple para cuantificar esto es calcular la diferencia entre la cantidad de nodos reales y la cantidad observada, así como para las aristas.

Como se observa en la figura 6.4, el exceso de nodos es muy importante y no es posible mejorarlo aumentando la cantidad de objetivos, por el contrario, empeora. Es especialmente preocupante que para una cantidad más que razonable de sondas (el 10 % de los nodos totales, por ejemplo), la topología final tenga un 200 % más de nodos que lo que debería.

6.4. Trabajo a futuro

El entorno de simulación desarrollado resultó ser muy versátil para el propósito original (familiarizarse con las herramientas y procedimientos), pero podía agregarsele muchas funcionalidades adicionales. Entre las más importantes destacan la utilización de MPI para realizar simulaciones distribuidas (y escalar a varios miles o millones de nodos), permitir conexiones

CSMA entre routers (con el apropiado método de resolución de aliases) y la implementación de un algoritmo eficiente y jerárquico para la asignación de direcciones IP. También a futuro es volver a realizar simulaciones con distintas topologías y estudiar como las mismas afectan los resultados encontrados: dejando fija la cantidad de nodos y variando el coeficiente de clustering, por ejemplo.

Capítulo 7

Pre exploración con RIPE

Habiendo realizado pruebas con un simulador, se procedió a trabajar con datos reales. Para esto, se utilizó la plataforma de medición RIPE ATLAS [43].

Hubo tres argumentos principales que motivaron esta decisión: a) teníamos créditos para realizar la exploración, b) soporta IPv6 y c) escala de manera simple para luego poder realizar exploraciones más grandes.

Dado que existen bastantes trabajos sobre IPv4, se decidió intentar descubrir una topología IPv6 a nivel IP. La elección de la resolución resulta un tanto forzosa: RIPE ATLAS no provee ningún mecanismo para realizar resolución de aliases y la implementación sobre su API resulta muy costosa (en cantidad de paquetes, que se traduce en créditos).

7.1. Objetivos y Metodología

El objetivo de la exploración fue comparar las topologías con IPv4 e IPv6 a nivel IP en América del Sur, en particular, diferencias de conectividad. En general es difícil poder hacer una compración entre dos topologías ya que los puntos de medición y los objetivos son distintos. Aprovechando que algunos nodos de RIPE ATLAS son dual-stack, se realizó una medición full-mesh (todos contra todos), utilizando como objetivos las dos direcciones IP de cada punto de medición (IPv4 e IPv6).

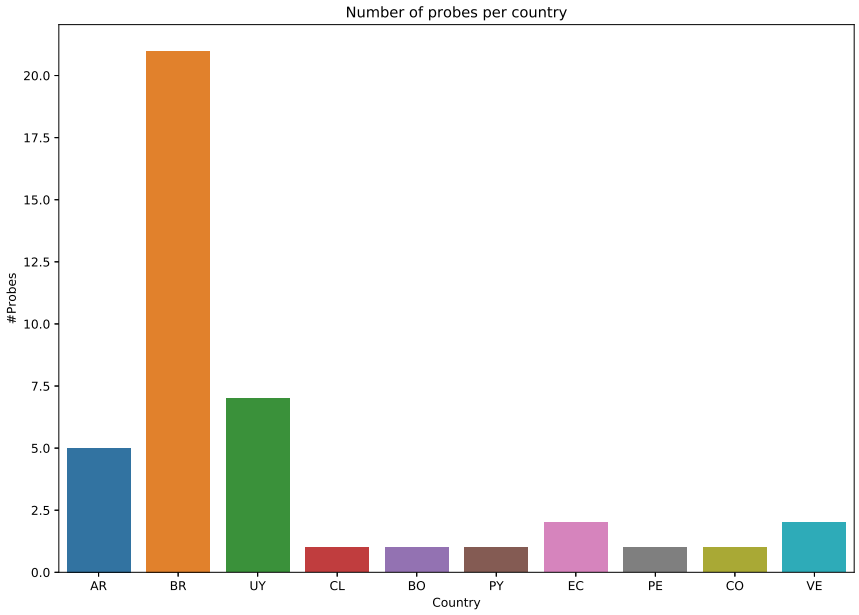


Figura 7.1: Distribución de las sondas entre los distintos paises para la pre exploración.

7.1.1. Elección de sondas

Las sondas de RIPE ATLAS pueden ejecutar varias versiones del sistema operativo. Dado que para las primeras versiones existen algunos problemas conocidos que afectan los resultados, se eligieron todas las sondas que operan con la versión 3.0. Aparte, RIPE ATLAS cuenta con un conjunto de tags que describen las capacidades de una sonda y su estado. En este sentido se verificó que las sondas pudiesen trabajar en IPv6. La Figura 7.1 muestra la distribución de las sondas elegidas en los distintos paises. De las 42 sondas seleccionadas originalmente, 39 respondieron a la hora de realizar la medición mientras que 3 no lo hicieron. Dos de las sondas que fallaron fueron en Brasil y una en Uruguay.

7.1.2. Elección de objetivos

Como objetivos se usaron las direcciones IPv4 e IPv6 de cada sonda. De los puntos de medición que fallaron, tampoco se puedo obtener sus direcciones IP, por lo que en total se usaron 78 objetivos.

7.1.3. Procesamiento de los datos

Existe poca literatura sobre como procesar el resultado de un traceroute. CAIDA [21] hace publico un método que no preserva la conectividad ya que desecha la respuesta del destino por ejemplo. Para este trabajo se utilizó el siguiente método:

1. Para cada salto, quedarse con la última respuesta que no sea un non-responsive.
2. Cada IP privada tratarla como non-responsive.
3. Agrupar non-responsives consecutivos como un solo non-responsive. Cada grupo obtiene un identificador distinto.
4. Inferir una arista para todo par de IPs (o non-responsive agrupado) consecutivas.

La desventaja del procedimiento es que agrega nodos extra y que necesariamente diferencia entre non-responsive cuando podría pasar que se trate del mismo.

Para una discusión posterior de los resultados se realizó una geolocalización a nivel de continente y de país. Para el nivel continental se utilizó la asignación de prefijos a los distintos RIRs combinado con la base de datos MaxMind [31]. Para la resolución de paises se utilizó unicamente MaxMind.

7.1.4. Resultados

Idealmente se deberían haber realizado 3042 traceroutes, en la práctica sin embargo, siempre algunos no se realizan. En total se recolectaron 2822 resultados. De los 2822: 1383 se corresponden a IPv4 y 1439 a IPv6. La medición tomó 40 minutos aproximadamente.

G	$\|V\|$	$\|E\|$	$\|*\|$	$\langle c \rangle$	$\langle k_{nn} \rangle$
G_{v4}	3986	7209	2478	0.017	0.074
G_{v6}	2851	5292	2007	0.019	0.117

Cuadro 7.1: Estadísticas de las dos topologías obtenidas. $|*|$ es la cantidad de nodos non-resonisve, $\overline{c}$ es el coeficiente de clustering promedio y el promedio del grado promedio de los vecinos $\overline{knn}$.

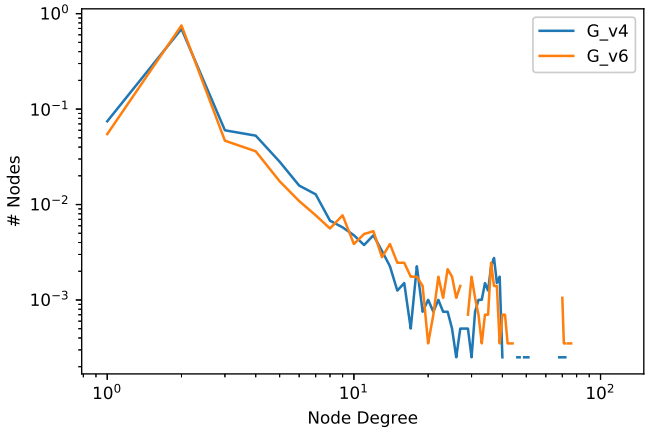


Figura 7.2: Distribución de grado para los grafos G_{v4} y G_{v6} en escala log-log.

7.2. Construcción de topologías

Utilizando la metodología descrita anteriormente se construyeron dos topologías: G_{v4} y G_{v6} con nodos correspondientes a IPv4 e IPv6 respectivamente. La Tabla 7.1 muestra las principales características de las topologías obtenidas. Comparando con [21] se puede ver que los coeficientes de clustering son prácticamente iguales mientras que $\overline{knn}$ da considerablemente más alto. Observando la distribución de grado (Figura 7.2) de ambos grafos se puede apreciar el efecto de cola pesada característico de toda red compleja.

De las estadísticas anteriores se desprende que las topologías son bastante similares, algo que en principio no era de esperar. Como veremos a continuación hay muchas posibles explicaciones para esto: la más probable parecería ser que la infraestructura utilizada en ambos casos es en gran medida la misma (gran presencia de routers dual-stack, por ejemplo).

7.3. Análisis de conectividad dentro de América Latina

No siempre es posible comparar conectividad entre IPv4 e IPv6. Para lograr extraer buenas conclusiones es necesario poder aislar otros factores como la variación de rutas en el tiempo, diferencias en los puntos de medición, etc.

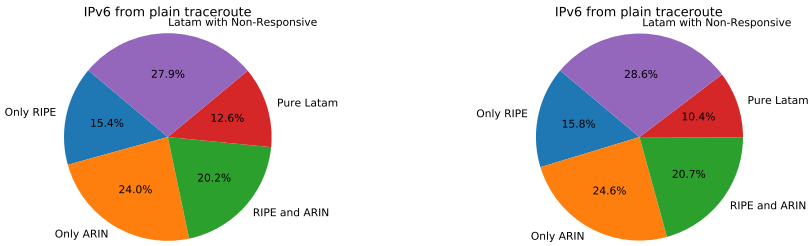
Dado que todos los nodos se encuentran en América Latina, se intentará estudiar que porción del trafico interno deja el continente y si existe alguna diferencia entre las dos versiones de IP.

7.3.1. Estadística global

La forma más fácil de análsis consiste en considerar solo los traceroutes originales. Luego, se asocia a cada nodo su RIR. Interesa diferenciar el caso entre LACNIC y no LACNIC.

Los antes mencionados tienen todos como origen y destino nodos en LACNIC por lo que interesa estudiar si en los nodos internos se detecta alguna dirección fuera del continente.

Figura 7.3: Clasificación de traceroutes en base a los RIRs atravesados.



(a) Distribución de caminos en IPv4 (b) Distribución de caminos en IPv6

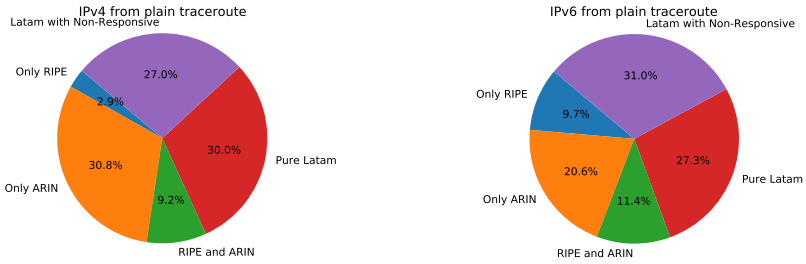
En la Figura 7.3 se ve que para ambas versiones del protocolo, un 60 % de los caminos observados abandona a ciencia cierta el continente. Por otra parte, el porcentaje que se corresponde a nodos non-responsive podría significar permanencia o no. En este sentido, trataremos a este tipo de nodos como el margen entre el mejor y el peor caso. En el primer escenario se corresponden con nodos en LACNIC y en el segundo no.

Lo que es interesante también observar es que el comportamiento para IPv4 tanto como para IPv6 es muy similar.

Utilizando la topología inferida

Para el análisis anterior no fue necesario el uso de las topologías inferidas. Se realizará un estudio similar, tomando en esta ocasión todos los caminos más cortos que tienen como origen y destino un nodo en LACNIC. A estos caminos se les llamará latentes, nombre que se explicará más adelante. Los resultados se pueden apreciar en la Figura 7.4. Aunque entre versiones del protocolo IP se asemejan, hay una diferencia con respecto a los traceroutes originales.

Figura 7.4: Clasificación de caminos latentes en base a los RIRs atravesados



(a) Distribución de caminos en IPv4 (b) Distribución de caminos en IPv6

Se podría argumentar de forma rápida que esto se debe a que en Internet, no se utiliza el camino más corto [51]. Sin embargo, podemos encontrar una explicación que justifica la diferencia entre ambos resultados. Lo que es más, el método, realizado con cuidado permitiría hallar cotas entre las diferencias, permitiendo el estudio del problema dual.

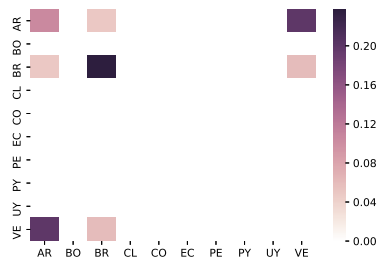
Considerese un traceroute genérico de 7 hops: $TR = (ip_1, ip_2, ip_3, ip_4, ip_5, ip_6, ip_7)$. Si TR es puro (LACNIC), entonces va a contribuir con $C_2^7 = 21$ caminos en el grafo. Por otra parte, si ip_4 está asignada a RIPE, habrá $2C_2^3 = 6$ nuevos caminos puros y $3 + 3 + 3 \times 3 = 15$ que abandonan el continente (los caminos que pasan por ip_4), por lo que todos traceroutes originales aportan caminos puros en el grafo. Puede pensarse como un caso discreto del teorema de conservación del signo.

7.3.2. Estadísticas por países

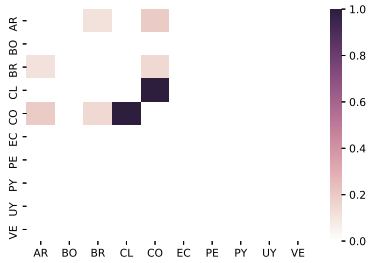
Para ganar un poco más de información sobre el comportamiento interno del trafico, se le asignó a cada IP el país de origen utilizando MaxMind [31] y luego se repitió el proceso realizado arriba: se estudió primero los traceroutes originales y luego los caminos latentes.

Para facilitar la visualización se construyeron los mapas de calor asociados a los caminos puros y puros con saltos non-responisve. Las Figuras 7.5, 7.6 muestran estos resultados.

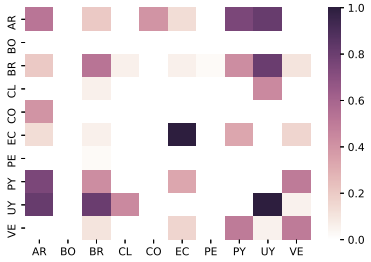
Figura 7.5: Fracción de traceroutes puros y con nodos non-responsive entre países de América del Sur.



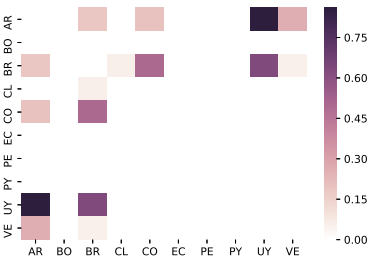
(a) LACNIC puro en IPv4



(b) LACNIC puro en IPv6



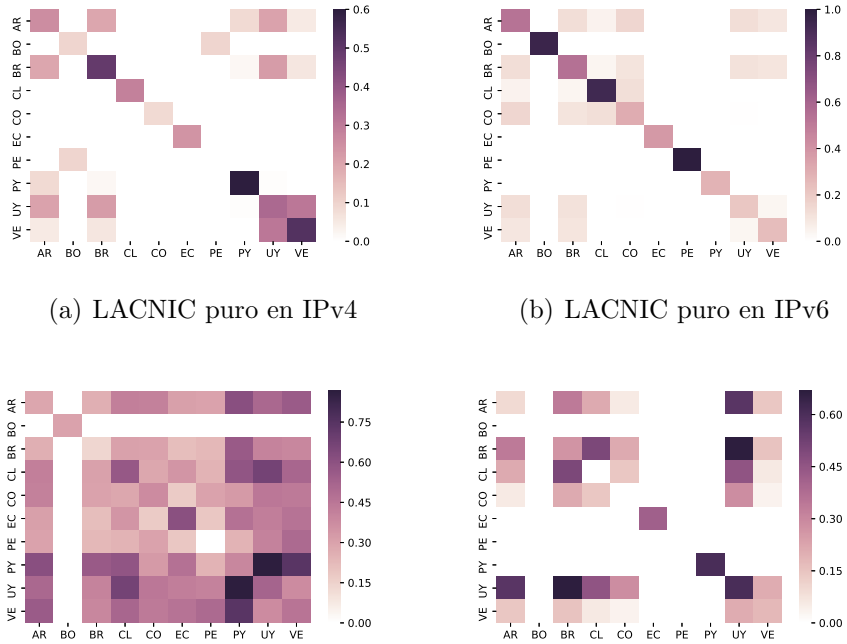
(c) LACNIC con Non-responsive en IPv4



(d) LACNIC con Non-responsive en IPv6

Es un poco más complejo comprar los traceroutes puros de los caminos latentes por la visualización, pero de observar con cuidado se tiene el mismo fenómeno que antes. Los caminos latentes presentan valores particularmente más altos en la diagonal, pero esto nuevamente es un efecto de la *conservación del signo*.

Figura 7.6: Fracción de caminos latentes puros y con nodos non-responsive entre países de América del Sur.



(a) LACNIC puro en IPv4

(b) LACNIC puro en IPv6

(c) LACNIC con Non-responsive en IPv4

(d) LACNIC con Non-responsive en IPv6

Se les llamó latentes pues por más que no pueden interpretarse necesariamente como el recorrido que haría un paquete en Internet, revelan la infraestructura que existe y los caminos que podría haber. Piensese en el siguiente escenario donde se midieron algunos traceroutes:

$TR1 = (ip_1,\ ip_2,\ ip_3,\ ip_4)$ donde ip_1, ip_2 se encuentran en Argentina y ip_3, ip_4 en Brasil. $TR2 = (ip_1,\ ip_2,\ ip_6)$ donde ip_6 está en Bolivia. Finalmente, supongase que Bolvia routea su tráfico hacia Brasil via ARIN y esto se observa por ejemplo en un tercer traceroute: $TR3 = (ip_6,\ ip_7,\ ip_8, ip_9,\ ip_{10},\ ip_3)$ donde $ip_i, i = 7, 8, 9, 10$ están asignadas a ARIN. En la topología a nivel IP, usando el camino más corto , se inferiría un camino entre Boliva y Brasil cuando este no existe. Por más que ese no es un camino real de los datos, podría llegar a serlo con un cambio en la política de tráfico. Si lo que se quiere es estudiar el desarrollo IPv6 en el continente, está es información valiosa.

7.4. Conclusiones

El estado de conectividad en América Latina aún no esta suficientemente desarrollado. El desglose por países muestra que por más que algunos países se interconectan de forma realtivamente buena, no es la norma. Por otra parte, sorprende la similitud en los resultados observados entre las dos versiones de IP. Era esperable que la diferencia fuese más marcada pero no fue el caso.

Para finalizar, mostramos como el estudio de los caminos más cortos en un grafo puede tener significancia a pesar de que estos no se corresponden con el camino de los datos en la red.

Capítulo 8

A la grande le puse cuca

Utilizando la misma plataforma que para la pre - exploración (Ripe Atlas), se realizó una campaña a gran escala en América Latina para descubir la topología a nivel IPv6. Todo el código pertinente a esta exploración así como el entorno para poder volver a reproducirla se encuentra en Github [1]

8.1. Objetivos y metodología

El objetivo de la exploración era obtener la topología a a nivel IPv6 más completa posible de América Latina. A diferencia de la exploración anterior no se pretendió realizar una comparación entre IPv4 e IPv6. Dado que los experimentos con Ripe Atlas son limitados, se optó por no realizar resolución de aliases, la cual es intensa en la cantidad de mensajes enviados. Existen muy pocas exploraciones a nivel IPv6 realizadas a gran escala. En particular nuestro objetivo consistió en lograr mejorar la topología IPv6 medida de forma periódica por CAIDA, en base a alguna métrica.

8.1.1. Elección de sondas

Dado que el objetivo es descubirir la topología a nivel de América Latina, la primera idea fue considerar todas y únicamente las sondas Ripe con capacidad IPv6 en América Latina. Al momento de la medición existían 43 sondas con estas características, sin embargo no se usaron todas por diversos motivos. El primero es que casi la mitad de ellas se encontraban en Brasil por

[1] `https://github.com/ellguso/topology_measurement`

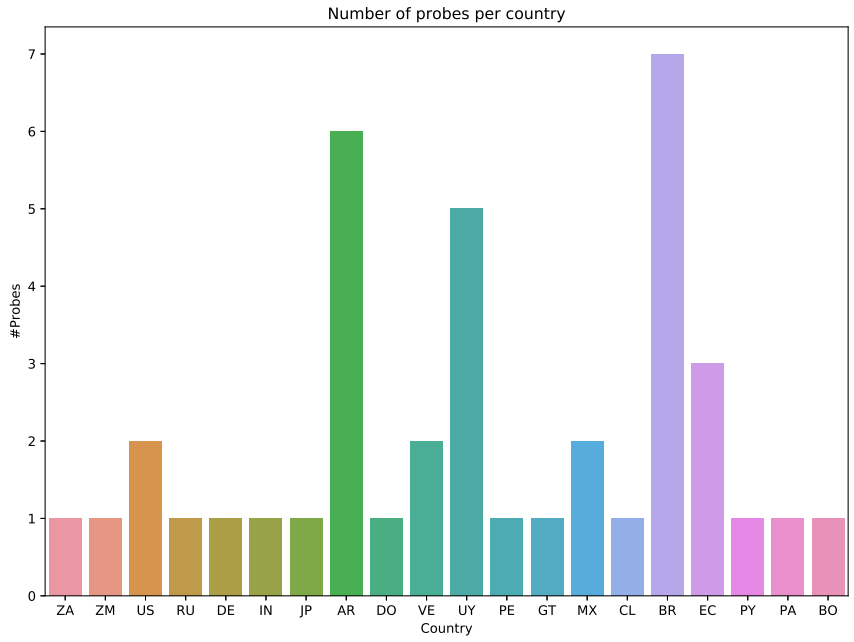


Figura 8.1: Distribución de las sondas usadas por país.

lo que había redundancia y sesgo en la distribución geográfica. El segundo es que se optó por elegir algunas sondas en otros continentes para diversificar la procedencia de los caminos y generar variabilidad en las aristas descubiertas. Por último cabe destacar que dada la restricción en la cantidad de créditos, no podíamos trabajar con más de 40 sondas.

La figura 8.1 muestra la distribución de las sondas por país.

8.1.2. Elección de objetivos

La elección de objetivos en IPv6, a diferencia de IPv4 es un problema abierto. En el viejo protocolo IP, es posible enviar traceroutes utilizando fuerza bruta a un conjunto inmenso de direcciones y lograr cubrirlas todas. En el proyecto Ark, por ejemplo, cada 3 días envían un traceroute a alguna dirección en cada /24 (2^{24}). IPv6 tiene dos problemas que IPv4 no: el espacio de direcciones es mucho más grande, por lo resulta imposible enviar

traceroutes a una dirección en cada /64 o /128 y la cantidad de direcciones en uso es despreciable comparado con las que no lo están. En base a los dos puntos anteriores es que se comenzaron a compaginar hitlists de direcciones IPv6 para usar como objetivos. La creación de hitlists es compleja puesto que muchas de las direcciones son dinámicas y al poco tiempo pierden validez. Gasser et.al. [17] confeccionaron una hitlist a partir de fuentes estáticas y la complementaron realizando traceroutes a estas direcciones para obtener nuevas. En el trabajo descrito en [15], mezclan el cálculo de la entropía de los nibbles en un conjunto de direcciones con redes bayesianas para predecir futuras direcciones, basados en un conjunto semilla inicial.

En esta exploración los requisitos eran más complejos pues se requería de forma adicional que todas las direcciones pertenecieran a América Latina. El criterio para decidir si una dirección pertenecía o no fue utilizar los prefijos asignados a LACNIC.

Para la creación de la hitlist utilizada se siguieron los siguientes pasos:

1. Listar todas las direcciones IPv6 en la versión más actualizada de la hitlist descrita en [17].

2. Listar todas las direcciones IPv6 encontradas en la exploración más reciente de CAIDA.

3. Filtrar todas las direcciones que no perteneciesen a: 2001:1200::/23 o 2800::/12.

4. Para cada /64 diferente, elegir una dirección al azar.

El último paso fue necesario al detectar que varias direcciones distintas pertenecían muchas veces al mismo servidor o granja de servidores, no aportando nuevos destinos reales para los traceroutes. El resultado final de este proceso fue una hitlist con 20741 direcciones IPv6.

8.1.3. Procesamiento de los datos

Dado que el objetivo es comparar los resultados con los obtenidos por CAIDA, se utilizó el proceso descrito en [21] para procesar los resultados.

8.1.4. Resultados

Es común que algunos de los traceroutes no se ejecuten. En la campaña se recolectaron resultados de 765396 de los 829640 (20741×40) que habían sido planificados ($92,2\,\%$).

La campaña demoró una semana. Esto se debe mayoritariamente a que la plataforma utilizada no permitía más de 100 medidas simultaneas.

Los resultados se encuentran publicados en [2]

8.2. Topologías inferidas

A partir de los datos recolectados se construyeron 3 topologías: G_{ip}, G_{ipl} y G_{as}. G_{ip} es la topología básica a nivel de IP obtenida al aplicar el procesamiento en forma estándar. G_{ipl} es el subgrafo inducido en G_{ip} de tomar solo los nodos que pertenecen a LACNIC o que son vecinos de un nodo en LACNIC. Por último, G_{as} es el grafo que resulta en asignarle a cada nodo en G_{ip} el sistema autónomo correspondiente. El Cuadro 8.1 resume las principales propiedades de cada una de estas topologías. La Figura 8.2 muestra una visualización de la topología inferida a nivel de AS utilizando la descomposición en k-cores.

Por otra parte, se obtuvieron los resultados de una campaña periódica realizada por CAIDA a nivel de IPv6 y se analizaron de la misma forma. Las características pueden observarse en el Cuadro 8.2.

Observando ambos cuadros hay algunas datos que llaman la atención. En particular, es interesante observar como a pesar de que los grafos CG_j son un orden de magnitud más grande que los G_j, propiedades como el clustering promedio o el exponente γ permanecen casi inalterados. En cierta medida, esto es un indicador que la topología encontrada en la campaña es *similar* a la global.

Otra peculiaridad es que los exponentes encontrados no se condicen con los valores que habían sido discutidos en la sección Modelos. Principalmente, esto se debe a los resultados anteriores estaban calculados para IPv4. Para IPv6 de hecho, las métricas conocidas son prácticamente nulas.

[2] `https://www.dropbox.com/s/sy1fgcvkl5bni28/cuca.gz?dl=0`

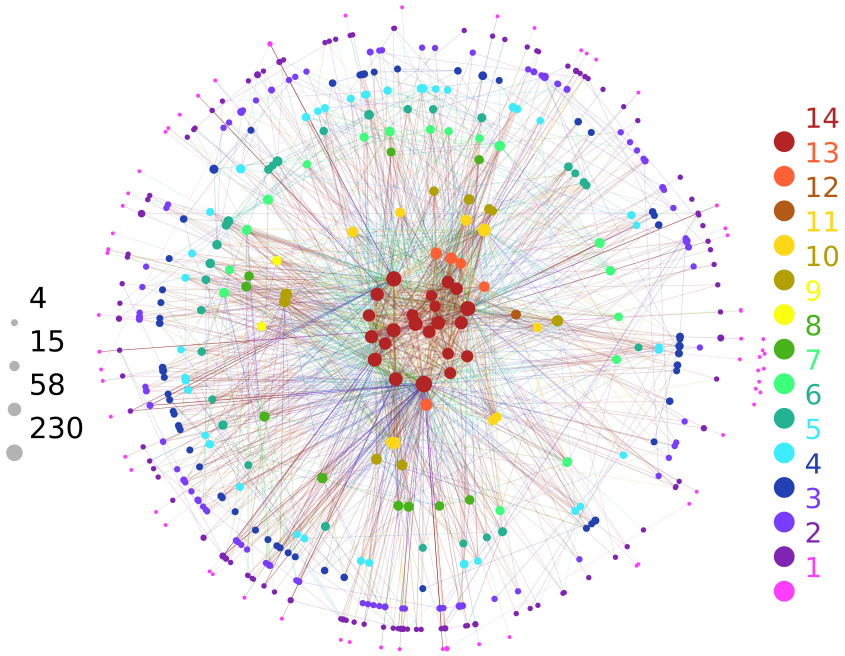


Figura 8.2: Visualización de la topología medida a nivel de AS usando la descomposición en k-cores. Visualización generada usando [27].

Cuadro 8.1: Principales características de las topologías inferidas.

	$\|V\|$	$\|E\|$	$\overline{c}$	γ
G_{ip}	18760	34695	0.02	2.99
G_{ipl}	15628	25326	0.02	3.04
G_{as}	604	2055	0.40	2.53

Cuadro 8.2: Características análogas a las calculadas para las topologías inferidas, observadas en una de las mediciones globales más recientes de CAIDA a nivel IPv6.

	$\|V\|$	$\|E\|$	$\overline{c}$	γ
CG_{ip}	202450	433236	0.01	2.58
CG_{ipl}	16930	26938	0.02	2.97
CG_{as}	7701	23845	0.37	2.73

Cuadro 8.3: Cantidad de aristas observadas en uno de los grafos o en ambos.

$G_{as} \backslash CG_{as}$	Sí	No
Sí	1023	445
No	1218	NC

8.3. Comparación de resultados

Se procede a analizar los resultados y a cuantificar el rendimiento de la campaña. Recordando que el objetivo era mejorar los resultados del proyecto Ark IPv6 en América Latina, una métrica que surge naturalmente es comparar la cantidad de aristas descubiertas en una campaña (entre nodos dentro del continente Americano) y no en la otra.

Dado que es poco probable observar la misma dirección IP y menos aún el mismo enlace, la discusión se hará a nivel de enlaces entre sistemas autónomos.

8.3.1. Metodología

Para obtener la métrica se siguieron los siguientes pasos:

1. Se obtiene del proyecto Routeviews todos los prefijos anunciados y su correspondiente número de AS.

2. Se genera una lista con todos los prefijos anteriores y su correspondiente ASN si están dentro de las subredes asociadas a LACNIC.

3. Para cada grafo [G_{as}, CG_{as}] se obtiene una lista de las aristas tal que ambos extremos se encuentran en la lista generada en el punto 2.

El Cuadro 8.3 resume la cantidad de aristas encontradas.

8.3.2. Análisis

De las 2686 aristas encontradas en total, nuestra campaña aportó un 17 % de enlaces nuevos. Si comparamos esto con el resultado de un trabajo como **IXPs Mapped?** y tenemos en cuenta nuestro objetivo, no es un mal resultado.

Cuadro 8.4: Tamaño de las campañas en cantidad de aristas por prefijos en América Latina.

	Media por prefijo	Desviación	Cant. Total
CAIDA	188.39	103.83	1210942
CUCA	3.14	55.41	20226

A pesar de esto, sería ingenuo asumir que todo salió como se esperaba o mejor: hay 1218 aristas que fueron descubiertas por CAIDA pero no por nuestra campaña.

Esto lleva a la pregunta: ¿será que por el diseño de nuestra campaña nos resultó imposible encontrar esas aristas?

Para contestar la pregunta anterior estudiamos que porcentaje de los prefijos totales en América Latina habían sido destino de algún traceroute. Parecería natural que si no hay paquetes destinados hacia algunos prefijos, no se descubran aristas allí.

Encontramos que CAIDA tiene destinos en el 99 % de los prefijos mientras que CUCA solo en el 56 %.

Idealmente, las 1218 aristas no encontradas tienen alguno de sus extremos en el 46 % de los prefijos que no fueron usados como destinos y eso explica no haberlas encontrado. Lamentablemente, solo 195 caen en está categoría. Las restantes podrían, en teoría, haber sido descubiertas.

Luego de un análisis cuidadoso de los datos, se encontró una explicación plausible a lo anterior. Dado que en Internet las rutas varían con el tiempo, se observa que la medición de un mismo trayecto repetidas veces a lo largo del tiempo resulta en nuevas aristas descubiertas. Esto significa que si una campaña envía muchos más traceroutes que otra, es natural que encuentre más aristas. La pregunta que resta contestar entonces es: ¿qué tanto más grande es la exploración de CAIDA que CUCA? La respuesta es mucho.

Para cuantificar lo anterior se calculó cuantos traceroutes eran dirigidos a cada uno de los prefijos asignados a LACNIC, aparte de naturalmente calcular la suma total de todos estos. En el Cuadro 8.4 se aprecia que la exploración realizada por CAIDA es dos ordenes de magnitud más grande que la nuestra. Esta es, finalmente, la explicación que justifica no haber encontrado algunas de las aristas.

Cuadro 8.5: Características de las topologías con datos de CUCA luego del cambio de metodología.

	$\|V\|$	$\|E\|$	$\overline{c}$	γ
G_{ip}	37905	165374	0.222225	2.407465
G_{ipl}	35625	157917	0.233024	2.381923
G_{as}	1386	11412	0.779170	2.552826

Cuadro 8.6: Características de las topologías con datos de CAIDA luego del cambio de metodología.

	$\|V\|$	$\|E\|$	$\overline{c}$	γ
CG_{ip}	4651387	5263146	0.001686	2.330830
CG_{ipl}	653691	719528	0.002035	16.059418
CG_{as}	13681	88903	0.683887	2.765026

8.4. Topologías inferidas: segunda vuelta

Luego del proceso anterior, se realizó una puesta en común de los resultados con Ignacio Alvarez-Hamelin quien sugirió algunas modificaciones al proceso anterior que describiremos a continuación.

Las diferencias mencionadas son a nivel del procesamiento inicial de los datos. Dado que esto cambia los resultados siguientes, también mostraremos los nuevos valores obtenidos.

El planteo es el siguiente: si la topología final que se desea obtener es a nivel de Ases, entonces no es tan grave inferir una arista entre dos nodos si existe uno o varios nodos non-responsive en el medio. Aparte, la IP de origen y destino no generan ruido y pueden ser incluidas.

El resultado de este cambio en la metodología es, trivialmente, una topología con más aristas. Los cuadros 8.6, 8.5 resumen las características de las nuevas topologías.

Se puede observar como a nivel de sistemas autónomos, ambas topologías duplican la cantidad de nodos y cuadriplican la cantidad de aristas. De alguna forma, el cambio que experimentan ambos grafos es similar. Sin embargo, los resultados realmente interesantes ocurren al observar la cantidad de aristas LATAM-LATAM descubiertas en ambas metodologías.

Se puede apreciar en el Cuadro 8.7 que bajo esta nueva óptica, nuestra campaña encuentra casi tres veces más aristas que el estado del arte. Es

Cuadro 8.7: Cantidad de aristas observadas en uno de los grafos o en ambos con la nueva metodología.

$G_{as} \backslash CG_{as}$	Sí	No
Sí	2703	7005
No	2700	NC

importante resaltar en este momento que el tamaño de la campaña sigue siendo el mismo, es decir que CAIDA sigue siendo dos ordenes de magnitud más grande que CUCA.

8.5. Análisis de latencia

También a sugerencia de Ignacio realizamos un estudio de la latencia entre países. Con este cometido, asignamos al origien y destino de cada traceroute realizado un país. Luego, considereamos todos los *rtt* en el último hop entre un origen y destino dado y tomamos el míninmo entre todos ellos. Los resultados se encuentran en la Figura 8.3.

Para la asignación de IPs a países tomamos un acercamiento similar al de la exploración preliminar pero más riguroso. Cada IP se consultó en MaxMind pero también en Team Cymru [52]. En caso de que ambas coincidieran o que una de las dos tuviese un país y la otra no, ese era el valor usando. En caso contrario, se descartaba la medición.

Observando con cuidado la Figura 8.3 hay algunas cosas que llaman la atención. Lo primero que hay que mencionar es que por simplicidad, todos los resultados fueron transformados en enteros. Luego, los -1 naturalmente son son valores reales, sino que denotan la ausencia de medición entre los valores considerados. Los valores más extraños son los 0 encontrados. Estos se corresponden con valores menores a 1 segundo. Esto parece razonable pero sorprende entre países como Brasil y Belice. Sin embargo, en un análisis cuidadosos se encuentra que existe un traceroute entre ambos países con un solo hop. La explicación más plausible para este tipo de comportamiento es la conexión entre ambos países a través de algún IXP.

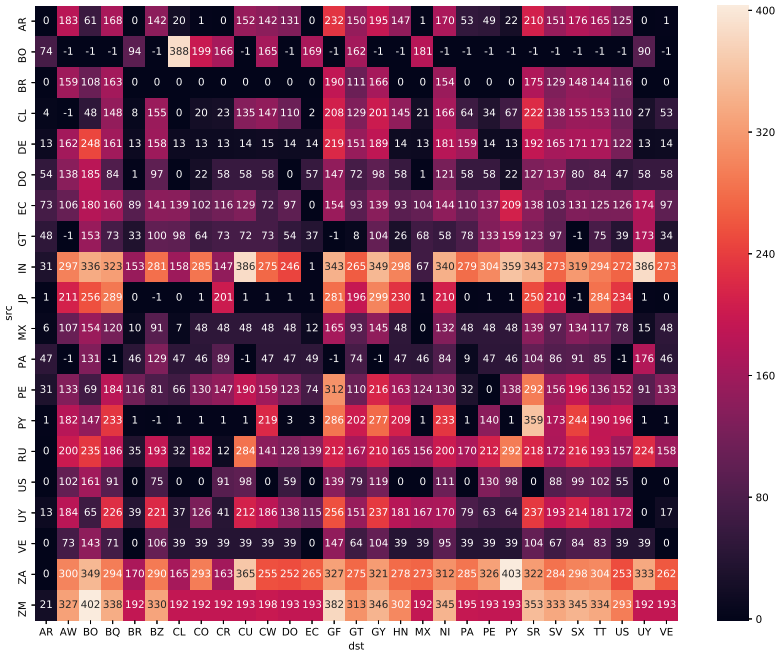


Figura 8.3: Heatmap de las distintas latencias entre países. Los -1 indican que no existe una medición con los origen y destinos indicados.

8.6. Consideraciones tecnológicas

Está última campaña presentó algunas dificultades tecnológicas más complejas que las anteriores, mayoritariamente por el volumen de datos utilizado.

Las implementaciones utilizadas en partes anteriores eran muchas veces naive. En este sentido, para procesar los traceroutes, estos se cargaban en memoria, se obtenían las aristas y se generaba el grafo. En este caso, sin embargo, el tamaño de los datos forzó a reescribir varios módulos.

Se mencionan a continuación algunos de los problemas encontrados y como fueron solucionados.

El análisis fue realizado en un servidor del INCO con 78GB de RAM y 8 cores.

8.6.1. Overhead en Python

Al intentar cargar en memoria los resultados de la exploración CUCA (3GB aproximadamente), se encontró que la lista no cabía en memoria. Es decir, que llegó a ocupar más de los 78GB disponibles. La solución fue reescribir los procedimientos que procesaban estos datos para trabajar como si fuese un stream, procesando de a un solo dato a la vez, escribiendo el resultado y liberando esa memoria.

8.6.2. Datos de CAIDA

Los resultados de su medición se encuentran formato warts y luego comprimidos. Existen varias bibliotecas para trabajar interactuar con el formato anterior. Sin embargo, la implementación en Python resultó extremadamente lenta y se tuvo que recurrir a su análoga en C. Los datos se extrajeron en formato CSV y dicho archivo pesaba 17GB. Fue necesario extaer la información relevante de ese archivo utilizando expresiones regulares, para las cuales nuevamente se trabajo línea a línea.

8.6.3. Resolución de ASN

Originalmente la resolución de ASN se hacía de forma naive utilizando la biblioteca **ipaddress**. Está cuenta con un procedimiento para verificar si una IP dada pertenece a una subred. El algoritmo tonto consistía para cada IP, recorrer la lista de prefijos (en el orden de 40000) y verificar si la IP dada se

encontraba o no. Esto en la práctica se tornó impracticable. Para solucionar este problema, se recurrió a la estructura de datos: *Merkel Patricia Trie* [34], un tipo de árbol optimizado para la busqueda por prefijos. Esta modificación redujo varios ordenes el tiempo de ejecución (de horas a segundos).

8.6.4. Ejecución de la campaña

Como se mencionó anteriormente, RIPE ATLAS permite únicamente 100 mediciones simultaneas (CUCA contaba con 20741). Dado que estar pendiente durante una semana del experimento para agendar nuevas mediciones era inviable, se implementó un pseudo planificador. Periódicamente, se consultaba cuantos experimentos había corriendo. De haber menos de 100, se agendaban los que se pudieran. Luego, se consultaba cuantos habían terminado y se removían de la plataforma (los experimentos a pesar de haber recolectado todos los datos permanecen activos a menos que se los detenga explícitamente). Para decidir si un experimento había terminado se uso la condición: ejecutó todos los traceroutes que tenía asignados o pasaron 30 minutos desde que comenzó. La justificación del criterio anterior es simple: cada medición intenta enviar 40 traceroutes desde cada uno de los puntos de medición. Si por alguna razón alguno de estos no está disponible, el experimento quedaría esperando indefinidamente en la plataforma, retrasando los que aún no fueron lanzados (recordar que puede haber solo 100 simultáneos).

Capítulo 9

Conclusiones

Para finalizar, resumimos las conclusiones del proyecto así como posibles líneas de trabajo a futuro.

9.1. Conclusiones

A lo largo de este proyecto se trabajó la construcción de topologías de Internet en sus diferentes niveles de resolución. Las mismas tienen dos componentes: uno técnico o tecnológico (como y que medir) y otro más bien teórico (como procesar y modelar).

En los primeros capítulos se resumió el estado del arte, explicando como mediante herramientas básicas utilizadas en las redes de computadoras se logra medir una topología. Se describieron diversas metodologías dependiendo del nivel de resolución deseado así como varios de los problemas que se presentan en la práctica. También aquí se explicaron las herramientas principales provenientes de Network Science como son los grafos aleatorios y las métricas definidas en estos. Luego de mencionar brevemente algunos proyectos de medición relevantes, se procedió a la experimentación de forma gradual.

En una primera instancia se implementó un entorno de trabajo en un simulador de eventos discretos sobre topologías sintéticas. A pesar de que el modelo usado no es el más realista ni la cantidad de nodos representativa, permitió ilustrar el proceso de resolución de aliases (y sus errores), anomalías en el comportamiento de routers, entre otros.

En las instancias siguientes se experimentó sobre la infraestructura real, con los problemas que ello conlleva. Los experimentos dejaron de ser ideales

para estar restrictos por limitaciones geográficas, de infraestructura, etc.

En este sentido, logramos realizar una exploración que bajo cierta metodología de procesamiento (la sugerida por Ignacio), supera ampliamente el estado del arte para América Latina.

De forma general, luego de dos experiencias midiendo IPv6, podemos afirmar que en comparación con IPv4, se sabe mucho menos y la documentación sobre como trabajar es bastante pobre. Virtualmente no hay métricas sobre las topologías medidas. Lo que es más, la forma de determinar los parámetros de las campañas (sondas, objetivos, etc) deja mucho que desar. En cuanto a IPv6 en América Latina, se encuentra poco desarrollado. En la primer campaña con RIPE encontramos que muy pocos países presentaban conectividad entre ellos y en la segunda que solo el 0,0005 % de las aristas posibles estaban presentes entre sistemas autonomos (junto a los resultados de CAIDA).

En cuanto al área en general, a pesar de haber comenzado hace casi 20 años, aún hay temas poco claros. Existe poca literatura que intente cuantificar los errores cometidos en un experimento dado, por ejemplo, para poder encontrar cotas en el error del coeficiente de clustering usadas. Lo que es más grave, prácticamente no existe un documento en el que se explique como procesar una traza para obtener la topología. En este trabajo intentamos seguir la breve descripción realizada por CAIDA en [21], pero aún esta es poco clara y no entre en detalles. Será necesario mejorar este aspecto para que la disciplina pueda seguir creciendo.

Relacionado a esto último, uno de los problemas más grandes a la hora de realizar experimentos es la reproducibilidad de los mismos. En Internet esto es más difícil aún pues está en constante cambio. En este trabajo contribuimos a mejorar este factor con dos entornos dockerizados: uno para realizar simulaciones y otro para medir utilizando Ripe Atlas, inferir y analizar topologías. La ventaja de estos entornos es que permiten volver a reproducir los mismos resultados, aparte de lograr un ambiente portable. Con respecto al como procesar las trazas, nuestro código es público por lo que podría ser usado como base para empezar a generar un estándar. Otro aporte que hacemos es publicar los resultados de nuestra campaña final, la primera en realizar una exploración regional en IPv6 (en América Latina).

Para finalizar, nos aceptaron un artículo en la SCCC 2017 en el que describimos la primera campaña (pre - exploración) y sus resultados.

9.2. Trabajo a futuro

Por lo que mencionamos antes, hay mucho que hacer aún en el área. En lo que respecta a lo realizado en este trabajo hay algunas líneas que parecen más razonables de seguir a futuro.

La primera es volver a repetir la última exploración intentando cubrir todos los prefijos en América Latina, aumentado también la cantidad de traceroutes enviados a cada uno de estos. Hoy en día no contamos con los recursos para esto, pero se podría intentar realizar un proyecto en conjunto con alguien que si los tenga.

Una segunda línea de trabajo es mejorar el entorno de simulación, logrando hacer funcionar *scamper*. Aparte, existe evidencia de ns3 distribuido, por lo que sería interesante ver si es posible extender el simulador para poder aumentar la cantidad de nodos y realizar experimentos más reales.

También sería interesante formalizar el proceso de medir Internet, definiendo procedimientos de acuerdo a las técnicas utilizadas, estableciendo como procesar los datos y logrando un marco común entre todos los investigadores. Para IPv6 en particular, seguirá siendo un problema abierto durante algún tiempo como definir el conjunto de objetivos.

Por último, hay muchos problemas teóricos de cuya solución se beneficiaría el área. En particular, estoy interesando en entender como una función que contrae nodos f sobre un grafo G afecta métricas como el coeficiente de clustering. Esto es lo que ocurre cuando se resuelven aliases o se asignan ASN. Encontrar cotas en las métricas resultantes podría servir para estudiar la topología solo a nivel IP, logrando así no introducir errores adicionales.

Apéndice A

Algunas deducciones para modelos matemáticos.

A.1. Erdös-Renyi

Sea $G_{N,p}$ el grafo Erdös-Renyi. Calculemos primero la cantidad esperada de aristas:$\langle E \rangle$.
La probabilidad de obtener m aristas puede calcularse de la siguiente forma: Supongase que las de las $\binom{N}{2}$ se eligen exactamente las aristas $e_1, e_2, \ldots, e_m$. La probabilidad de que todas ellas estén en el grafo es p^m, aparte la probabilidad de que ninguna de las otras esté es $(1-p)^{N-m}$. Por la regla del producto, la probabilidad de que un grafo específico con m aristas ocurra es $p^m(1-p)^{N-m}$. Hay $\binom{\binom{N}{2}}{m}$ formas de elegir las m aristas y por lo tanto esa cantidad distinta de grafos con m aristas.
Finalmente, la probabilidad de que un grafo tenga m aristas es $P(m) = \binom{\binom{N}{2}}{m} p^m (1-p)^{N-m}$. $P(m)$ sigue una distribución bionomial y es un resultado conocido que el valor esperado $\langle E \rangle = \sum mP(m) = \binom{N}{2} p$

Dado que cada arista contribuye a dos grados en la suma de todos los vértices, el grado promedio esperado ($\langle k \rangle$) es:

$$\langle k \rangle = \frac{2\langle E \rangle}{N} = p(N-1) \approx Np$$

cuando N es lo suficientemente grande.

Razonando de la misma forma que para las aristas, para que un vértice tenga grado k debo elegir k de los N vértices ($\binom{N}{k}$), con una probabilidad

p^k y que no tenga una arista con ningún otro $(1-p)^{N-k}$ llegando de esta forma a la expresión:

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}$$

nuevamente, es un resultado conocido que una distribución binomial, en el límite puede aproximarse por una Poisson de forma que reescribimos:

$$P(k) = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$$

Por último, consideremos todas las ternas $a-b-c$ donde a, b, c son nodos del grafo y a, b tanto como b, c son vecinos. Como las aristas son i.i.d, si a, b y b, c son vecinos la probabilidad de que a, c lo sean es p. El coeficiente de clustering se define exactamente como el cociente entre las cantidad de triángulos sobre la cantidad de triángulos no necesariamente cerrados y en este caso es p. Por lo tanto:

$$\langle c \rangle = p = \frac{\langle k \rangle}{N}$$

donde usamos el resultado anterior.

A.2. Barabási - Albert

La ecuación:

$$\frac{\delta k_s(t)}{\delta t} = \frac{m k_s(t)}{2mt + 2m_0 \langle k \rangle_0} \tag{A.1}$$

es una ecuación diferencial de variables separables por lo que podemos resolverla y obtener (usamos que $k_s(s) = m$):

$$\int_s^t \frac{\delta k_s(t)}{k_s(t)} = \int_s^t \frac{m \delta t}{2mt + 2m_0 \langle k \rangle_0} \tag{A.2}$$

$$Ln k_s(t) - Ln k_s(s) = \frac{1}{2}(Ln(2mt + 2m_0 \langle k \rangle_0) - Ln(2ms + 2m_0 \langle k \rangle_0)) \tag{A.3}$$

$$Ln(\frac{k_s(t)}{m}) = Ln \left(\frac{2mt + 2m_0 \langle k \rangle_0}{2ms + 2m_0 \langle k \rangle_0} \right)^{\frac{1}{2}} \tag{A.4}$$

$$k_s(t) = m\left(\frac{2mt + 2m_0\langle k\rangle_0}{2ms + 2m_0\langle k\rangle_0}\right)^{\frac{1}{2}} \tag{A.5}$$

Finalmente, suponiendo que $s, t \gg m_0\langle k\rangle_0$ se obtiene

$$k_s(t) = m\left(\frac{t}{s}\right)^{\frac{1}{2}} \tag{A.6}$$

Ahora, la distribución de probabilidad en un tiempo t podemos calcularla como:

$$P(k,t) = \frac{1}{t+m_0}\int_0^t \delta(k - k_s(t))ds \tag{A.7}$$

donde δ es la función Delta de Dirac. Lo que estamos haciendo es promediando los valores de $k_s(t)$ con valor k. La solución de esta integral es

$$P(k,t) = -\frac{1}{t+m_0}\left(\frac{\delta k_s(t)}{\delta s}\right)^{-1}\Big|_{s=s(k,t)}$$

donde $s = s(k,t)$ es la solución de la ecuación implícita $k = k_s(t)$.
Calculamos primero la derivada:

$$\frac{\delta k_s(t)}{\delta s} = m\frac{1}{2}\left(\frac{s}{t}\right)^{\frac{1}{2}}(-\frac{t}{s^2}) = -\frac{m}{2}\frac{t^{\frac{1}{2}}}{s^{\frac{3}{2}}}$$

luego resolvemos la ecuación implícita:

$$k_s(t) = k \iff \left(\frac{t}{s}\right)^{\frac{1}{2}} = k \iff \frac{k^2}{m^2} = \frac{t}{s} \iff s = \frac{tm^2}{k^2}$$

juntando todo:

$$P(k,t) = \frac{1}{t+m_0}\frac{2}{m}\frac{\left(\frac{tm^2}{k^2}\right)^{\frac{3}{2}}}{t^{\frac{1}{2}}} = \frac{2m^2tk^{-3}}{t+m_0}$$

la distribución que nos interesa es tomando el límite $t \to \infty$ donde se *cancelan* los t y obtenemos la distribución deseada:

$$P(k) = 2m^2k^{-3} \tag{A.8}$$

Apéndice B

Un último modelo con aplicaciones

B.1. Introducción

En recientes trabajos se ha observado que si se considera Internet como un grafo embebido en un espacio métrico, entonces la curvatura del mismo es negativa y modelos basados en la geometría hiperbólica serían más aptos (en contraste con la euclidiana como el modelo HOT) [48, 47].

A partir de eso, y con raíces en el modelo de Conexión Preferencial, Kurikov et.al. [40] propusieron un modelo que llaman PSO (Popularity x Similarity Optimization) que hace gran uso de la geometría latente del problema.

En [39] Kurikov et.al proponen un algoritmo: HyperMap para embeber el grafo obtenido en un plano hiperbólico. Una de las aplicaciones que tiene esto es que les permite predecir con gran precisión aristas faltantes.

Con el fin de mostrar este modelo y sus resultados primero haremos una breve introducción a la geometría hiperbólica.

B.2. Geometría hiperbólica

Geometría hiperbólica (GH) o geometría de curvatura negativa constante es un área extensa de la matemática y en esta sección no se pretenderá ser dar una explicación extensiva de la misma sino un breve resumen de lo mínimo necesario para entender el modelos siguientes.

La geometría clásica o euclidiana se define de forma axiomática. El primero en realizar esto fue Euclides con cinco axiomas. El quinto postulado establece que existe una única recta paralela a otra por un punto dado (este no fue el postulado original de Euclides, pero es equivalente). Existieron muchísimos esfuerzos por probar este postulado a partir de los otros cuatro sin éxito. Si en lugar de exigir que exista una única paralela se admiten infinitas se obtiene una geometría consistente, la hiperbólica.

Axiomas de Euclides

Axioma 1 Dados dos puntos se pueden trazar una recta que los une.

Axioma 2 Cualquier segmento puede ser prolongado de forma continua en una recta ilimitada en la misma dirección.

Axioma 3 Se puede trazar una circunferencia de centro en cualquier punto y radio cualquiera.

Axioma 4 Todos los ángulos rectos son iguales.

Axioma 5 Si una recta corta a otras dos formando, a un mismo lado de la secante, dos ángulos internos agudos, esas dos rectas prolongadas indefinidamente se cortan del lado en el que están dichos ángulos.

Axioma 5' Por un punto exterior a una recta, se puede trazar una única paralela a la recta dada.

B.2.1. Modelo del plano superior

Para comenzar a formalizar la GH consideraremos el modelo del plano superior (el borde no está incluido y representa el infinito):

$$\mathbb{H}^2 = \{z \in \mathbb{C},\ Im(z) > 0\} \tag{B.1}$$

$$\partial\mathbb{H} = \{z \in \mathbb{C} | Im(z) = 0\} \tag{B.2}$$

Explicaremos como se calcula la distancia entre dos puntos en $\mathbb{H}^2$. Una explicación más detallada se encuentra en el apéndice C. Si tenemos una curva

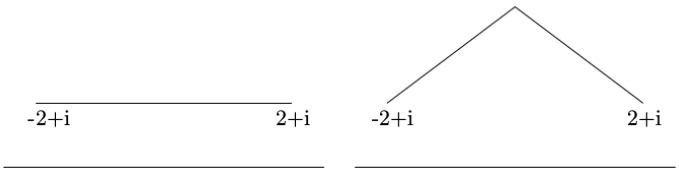


Figura B.1: Dos curvas entre los puntos $-2+i$ y $2+i$. Usando la fórmula de distancia hiperbólica, la de la izquierda mide 4 y la de la derecha 3,1. Figura extraída de [22]

en forma paramétrica $\gamma(t) = (x(t), y(t))$ y dotamos a $\mathbb{H}^2$ con la métrica dada por la ecuación B.3, en lugar de la clásica ($ds^2 = dx^2 + dy^2$)

$$ds = \frac{dx^2 + dy^2}{y^2} \tag{B.3}$$

podemos calcular el largo de un arco hiperbólico $\gamma\colon [a, b] \to \mathbb{H}$ como:

$$length_{\mathbb{H}}(\gamma) = \int_\gamma \frac{1}{Im(z)} dz = \int_a^b \frac{|\gamma'(t)|}{Im(\gamma(t))} dt \tag{B.4}$$

y finalmente definir la distancia entre dos puntos $z_1, z_2 \in \mathbb{H}^2$ como:

$$d(z_1, z_2) = \text{ínf}\,\{length_{\mathbb{H}}(\gamma) | \gamma \in X\} \tag{B.5}$$

donde X es el conjunto de todas las curvas diferenciables a trozos entre z_1 y z_2.

La curva γ en la que se alcanza el mínimo de la ecuación B.5 se llama geodésica.

A diferencia de $\mathbb{R}^2$ el camino más corto entre dos puntos (z_1 y z_2) no siempre se corresponde con una línea recta. Se tiene dos casos:

- Si $Re(z_1) = Re(z_2)$ el camino más corto es la segmento vertical que los une.
- Si $Re(z_1) \neq Re(z_2)$ el camino más corto entre ellos es el único arco de circunferencia que pasa por z_1, z_2 y que incide perpendicularmente en el eje real.

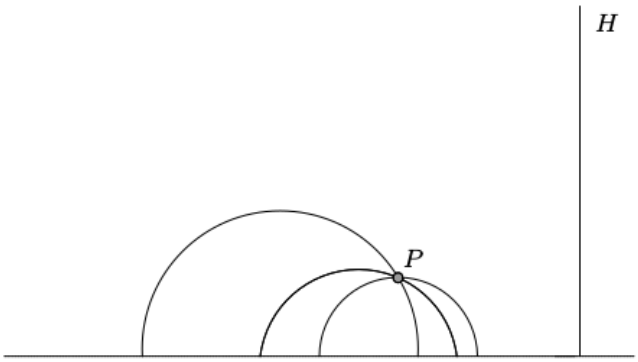


Figura B.2: Infinitas geodésicas paralelas a una recta dada que pasan por el punto P. Figura extraída de [22]

Tal como se definió el modelo, estas son todas las geodésicas y son únicas. Otra ventaja de la métrica elegida es que los círculos hiperbólicos son círculos euclídeos pero con distinto centro.

B.2.2. Disco de Poincaré

El modelo del plano superior tiene sus virtudes, pero no es el utilizado en los modelos de interés, por lo que presentaremos el modelo del Disco de Poincaré. Para esto, definiremos una biyección $h\colon \mathbb{H} \cup \partial\mathbb{H} \to \mathbb{D} \cup \partial\mathbb{D}$ donde:

$$\mathbb{D} = \{z \in \mathbb{C} || z| < 1\} \tag{B.6}$$

$$\partial\mathbb{D} = \{z \in \mathbb{C} || z| = 1\} \tag{B.7}$$

$$h(z) = \frac{z - i}{iz - 1} \tag{B.8}$$

Dado que h preserva ángulos (transformación conforme) y transforma bordes en bordes, se prueba que las geodésicas en $\mathbb{D}$ son los diámetros y los arcos de circunferencia que inciden de forma ortogonal en $\partial\mathbb{D}$. Aparte, h es una isometría por lo que $d_{\mathbb{D}}(h(z_1), h(z_2)) = d_{\mathbb{H}}(z_1, z_2)$ y nuevamente, los círculos hiperbólicos se corresponden con círculos euclídeos con centros desfasados.

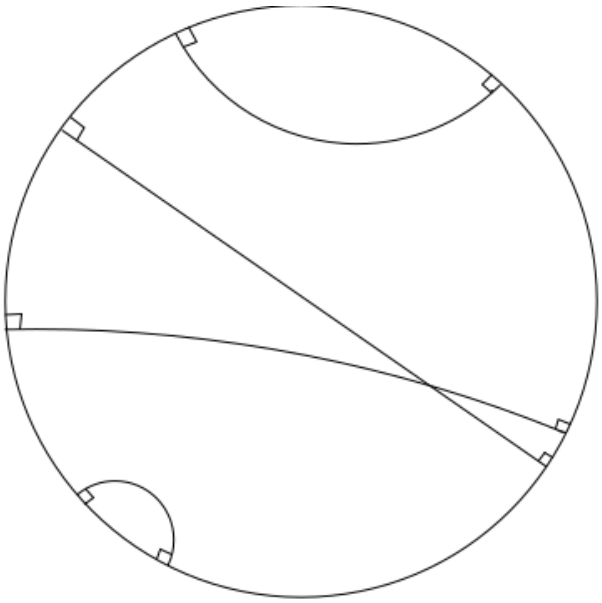


Figura B.3: Geodésicas en $\mathbb{D}$.

Una alternativa para calcular la distancia entre dos puntos en $\mathbb{D}$ que resultará útil en secciones posteriores es utilizando el teorema del coseno hiperbólico. Sean dos puntos $z_1, z_2 \in \mathbb{D}$ en coordenadas polares (r_i, θ_i), la ecuación para la distancia entre ellos (x_{ij}) es:

$$x_{ij} = \frac{1}{2} arccosh(\cosh\ 2r_1 \cosh\ 2r_2 - \sinh\ 2r_1 \sinh\ 2r_2 \cos\ \theta_{12}) \qquad \text{(B.9)}$$

B.3. El modelo PSO

Recordando un concepto manejado a lo largo de este trabajo, una de las principales características buscadas en un modelo es que la distribución de grado presente una cola pesada o **power-law**. Como se ha discutido previamente, modelos basados en el acoplamiento preferencial logran producir las distribuciones de grado buscadas a costa de un bajo coeficiente de clustering. Lo que es más, podría objetarse que no es un modelo representativo de las

Figura B.4: Circle Limit IV, M.C. Escher. Todos los demonios tienen el mismo tamaño y son deformados por la representación en $\mathbb{D}$.

dinámicas reales ya que no contempla conexiones entre pequeños nodos (peer to peer links).

Papadopoulos et.al propusieron en [40] un modelo que combina la concepto original del acoplamiento preferencial (al que llaman popularidad) con la idea de que existen conexiones entre nodos pequeños pero con características comunes (al que llaman similaridad). Este capítulo culmina en el estudio de este modelo, el PSO (Popularity versus Similarity Optimization). Este no solo tiene una precisión muy alta al analizar la topología a nivel de ASes, sino que también presenta un alto poder predictivo, el que podría usarse para inferir enlaces no descubiertos en una campaña de medición.

En [40] se presentan diversas variantes del modelo PSO, la más simple de todas siendo la descrita en el Algoritmo 1. En el mismo, la distancia angular es la medida de similitud y el tiempo de nacimiento (i en el Algoritmo 1) la de popularidad. Dado que se busca minimizar el producto de ambas, los primeros nodos (con menor tiempo de nacimiento son preferidos) antes que nodos con mucha distancia angular.

Algorithm 1 PSO Básico

Input: $m > 0$, $N > 0$
  $G = (V, E), V = \emptyset, E = \emptyset$
  for $i = 0$ **to** N **do**
    $\theta_i \leftarrow rand([0, 2\pi)),\ v_i \leftarrow (i, \theta_i)$
    $V \leftarrow V \cup v_i$
    if $t \leq m$ **then**
      $E \leftarrow E \cup \{(v_1, v_i), \ldots, (v_{i-1}, v_i)\}$
    else
      $E \leftarrow E \cup \{(b_1, v_i), \ldots, (b_m, v_i)\}$ donde $\{b_i\}_{i \leq N}$ son los nodos v_j que minimizan $j \times \widehat{v_j v_i},\ j < i$
    end if

  end for

Podemos dar una interpretación geométrica al Algoritmo 1. Si al tiempo de nacimiento le asignamos el valor $r_i = \log(i)$, entonces podemos ubicar al vértice v_i con coordenadas polares (r_i, θ_i) en el plano. La ventaja de esto es que ahora minimizar el producto $j \times \widehat{v_j v_i}$ es equivalente a minimizar la distancia hiperbólica $d_{\mathbb{H}}(v_i, v_j)$ (esto ocurre pues la fórmula para la distancia deducida antes es aproximadamente $r_i + r_j + ln(\frac{\theta_{ij}}{2})$

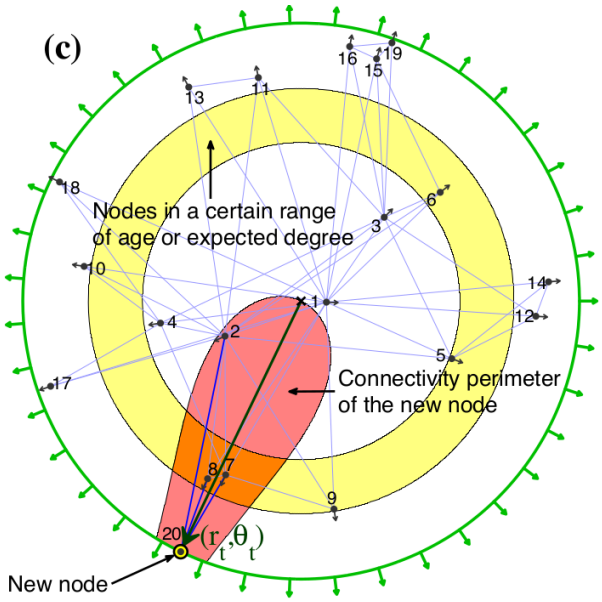


Figura B.5: Interpretación geométrica del algoritmo PSO con m=3, simulado hasta $i = 20$. El área rosada se corresponde con los nodos a distancia menor que r_t del nodo 20. El dibujo se encuentra a escala euclídea, por lo que el circulo hiperbólico se deforma al representarse. Figura extraída de [40].

Teniendo en cuenta que en la realidad se observa un declive de la popularidad con el tiempo, se agrega una variación al algoritmo que refleje este fenómeno.

Recordando que la popularidad depende de la distancia al centro, disminuir la popularidad se logra actualizando el radio del nodo i en el tiempo t de acuerdo a la ecuación B.10

$$r_i(t) = \beta r_i + (1-\beta) r_t,\ \beta \in [0,1],\ r_j = \ln(j) \tag{B.10}$$

Lo que es más, para agregar versatilidad al modelo, en lugar de que un nuevo nodo se conecte a los m nodos más cercanos, para cada nodo ya existente, estos se conectan con una probabilidad dada por la ecuación B.11.

$$p(x_{ij}) = [1 + e^{\frac{x_{ij}-R_j}{T}}]^{-1},\ R_j = r_j - 2\log\left(\frac{1}{\sin \pi T}\frac{1-e^{-(1-\beta)r_t}}{m(1-\beta)}\right) \tag{B.11}$$

En el nuevo esquema, $\beta \in [0,1]$ controla la degradación de la popularidad con el tiempo y es el que determina el exponente de la ley de potencias $P(k) \sim k^{-\gamma},\ \beta = \frac{1}{\gamma - 1}$. $T \in [0,1)$, la temperatura, regula el coeficiente de clustering en la red. Por último m es el grado esperado de cada vértice. Teniendo todo esto en cuenta el nuevo modelo está descrito por el Algoritmo 2:

Algorithm 2 PSO Final

Input: $m > 0$, $N > 0$, $\beta \in [0,1]$, $T \in [0,1)$

```
G = (V, E), V = ∅, E = ∅
for i = 0 to N do
    for v ∈ V do
        Actualizar el radio de acuerdo a la ecuación B.10
    end for
    θ_i ← rand([0, 2π)), v_i ← (i, θ_i)
    V ← V ∪ v_i
    for j = 0 to j = i − 1 do
        E ← E ∪ (v_i, v_j) con probabilidad dada por la ecuación B.11
    end for
end for
```

Se puede probar que la distribución de grado producida por el modelo en cuestión es idéntica a la del modelo de Barabasi-Albert. Sin embargo, a

diferencia del primero, el clustering no se desvanece conforme crece N. Como se mostrará a continuación las simulaciones producen un valor muy similar al observado en redes reales.

La última variante que se considerará del modelo es la inclusión de aristas internas. Es decir, que en un tiempo t, podrán aparecer aristas entre nodos ya existentes. El algoritmo 3 describe el modelo PSO-Generalizado.

Algorithm 3 PSO Generalizado

Input: $m > 0$, $N > 0$, $\beta \in [0, 1]$, $T \in [0, 1)$, $L > 0$
  $G = (V, E), V = \emptyset, E = \emptyset$
  for $i = 0$ **to** N **do**
    for $v \in V$ **do**
      Actualizar el radio de acuerdo a la ecuación B.10
    end for
    $\theta_i \leftarrow rand([0, 2\pi)),\ v_i \leftarrow (i, \theta_i)$
    $V \leftarrow V \cup v_i$
    for $j = 0$ **to** $j = i - 1$ **do**
      $E \leftarrow E \cup (v_i, v_j)$ con probabilidad dada por la ecuación B.11
    end for
    repeat
      $E \leftarrow E \cup (v_k, v_j)$ con $j, k < i$ elegidos al azar
    until Se agregan L nuevas aristas
  end for

B.3.1. HyperMap

En [40] así como en [39] se propone una metodología para embeber una topología dada en el modelo PSO. Se detallará el modelo HyperMap y sus resultados al trabajar con una topología a nivel de ASes.

La metodología detrás de HyperMap es simple, si el orden de aparición de los vértices es conocido, entonces también lo es su popularidad (radio), por lo que para poder realizar el embedding el único dato a calcular es la coordenada radial. Si se supiesen las coordenadas de todos los nodos, entonces sería posible calcular la probabilidad de que el grafo resultante sea el observado, por lo que aplicando la regla de Bayes es posible calcular la probabilidad condicional reversa. La ecuación B.12 es el resultado de realizar dichos cálculos. En el algoritmo 4, $\zeta = \sqrt{-K}$ donde K es la curvatura del

plano hiperbólico que se agrega como parámetro por generalidad (aunque no afecta el algoritmo). Existe otro pequeño cambio con respecto al algoritmo 3. La aparición de enlaces internos evita la reconstrucción hacia atrás de las aristas, por lo que HyperMap trabaja con una variante equivalente, en lugar de agregar enlaces internos y externos regulados por los parámetros m y L, la cantidad de conexiones que establece el nodo i en el tiempo t está dada por $m_i(t) = m + \overline{L_{i,t}(L)}$ donde $\overline{L_{i,t}(L)}$ es la cantidad esperada de enlaces internos entre i y los nodos ya existentes: $R_i = r_i - \frac{2}{\zeta} \log\left(\frac{2T}{\sin \pi T} \frac{1-i^{-(1-\beta)}}{m_i(t)(1-\beta)}\right)$

Algorithm 4 HyperMap, extraído de [38]

Input: $G = (V, E)$

Ordenar los vértices por grado de forma decreciente $k_1 > k_2 > \ldots k_N$

Se nombra al nodo con grado k_i: i.

Nace el nodo 1 y con radio $r_1 = 0$ y coordenada angular θ_1 aleatoria en $[0, 2\pi]$

for $i = 2$ **to** $i = N$ **do**

Nace el nodo y con radio $r_i = \frac{2}{\zeta} \ln(i)$

Se actualiza el radio de los nodos ya existentes de acuerdo a la ecuación B.10

Asignarle al nodo i coordinada radial θ_i tal que máxima $\mathbb{L}_2^i$ dado por la ecuación B.12

end for

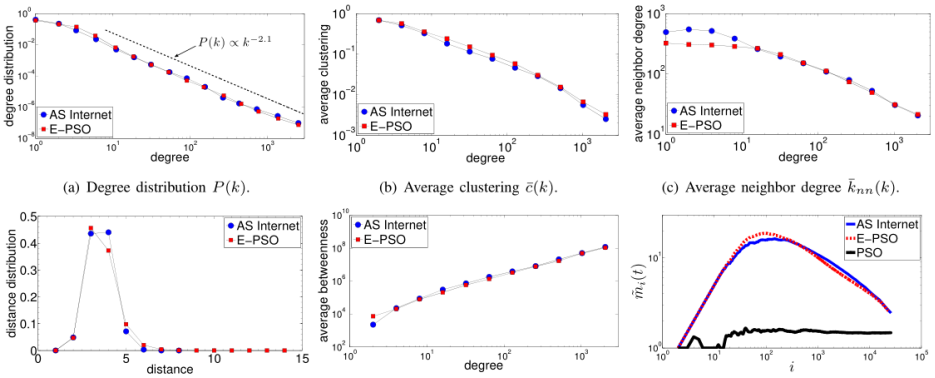
$$\mathbb{L}_2^i = \prod_{1 \leq j < i} p(x_{ij})^{\alpha_{ij}} [1 - p(x_{ij})]^{1-\alpha_{ij}} \tag{B.12}$$

B.3.2. Resultados

Se muestran a continuación los resultados presentados en [38] en los que se aprecia la efectividad del modelo para topologías de Internet. Pueden dividirse en dos tipos, los que involucran topologías sintéticas y los que embeben una topología real.

Papadopoulos et.al trabajaron con una topología a nivel de ASes recolectada por CAIDA en Diciembre del 2009 que se encuentra disponible en [23]. La topología es obtenida primero mediante una campaña de traceroutes (Ark) y luego usando RouteViews para determinar a que AS corresponde cada dirección IP.

Figura B.6: Figura extraída de [38]



(a) Degree distribution $P(k)$. (b) Average clustering $\bar{c}(k)$. (c) Average neighbor degree $\bar{k}_{nn}(k)$.

(d) Distance distribution $d(l)$. (e) Average betweenness $\bar{B}(k)$. (f) $\tilde{m}_i(t)$ vs. node appearance time i.

Fig. 1. Properties of the AS Internet vs. simulated networks grown according to the E-PSO model.

Por otra parte, el modelo requiere de datos históricos, como la cantidad inicial de enlaces que tiene un nuevo AS. Para obtener estos datos, utilizaron el data set histórico descrito en [11]. Este, a diferencia del anterior utiliza los dumps de RouteViews y RIPE, restricto a enlaces de tipo p2c ya que no cuentan con información suficiente p2p. A partir de estos dos datasets logran obtener los parámetros necesarios para simular usando E-PSO (la versión adaptada a HyperMap de la versión generalizada de PSO). La topología cuenta con $N = 25910$ ASes, exponente $\gamma = 2{,}1$, grado promedio $\overline{k} = 5$ y clustering promedio $\overline{c} = 0{,}61$. En base a los datos históricos se observa que la cantidad de conexiones iniciales es $m \approx 1{,}5$, de lo que se puede también calcular $L = \frac{\overline{k}-2m}{m} \approx 1$. β se elige en función de γ, por lo que: $\beta = 1 + \frac{1}{2{,}1} \approx 1{,}48$ y se encuentra que empíricamente que $T = 0{,}45$ para que $\overline{c} = 0{,}61$.

Con todos los parámetros definidos se simula la topología sintética y se comprara con la extraída de CAIDA en 2009. La figura B.3.2 muestra los resultados obtenidos.

Se puede observar que las métricas más importantes son virtualmente idénticas en la red sintética y la red real. Esto parecería indicar que una simulación a $N = 50000$ sería una buena aproximación de como se verá Internet cuando tenga 50000 Sistemas Autónomos.

Como ha sido mencionado anteriormente, una de las características más impresionantes del modelo es su capacidad para predecir aristas. Para medir la precisión de las predicciones Papadopoulos et.al. utilizaron la métrica Area

Figura B.7: La columna de la izquierda considera todos los posibles pares de nodos mientras que la segunda solo aquellos que no tienen nodos en común; en dicho caso el poder predictivo de los otros métodos es considerablemente peor. CN = Common-Neighbors, DP = Degree-Product, ISP = Inverse-Shortest-Path, Katz = Katz Index, HRG = Hierarchical Random Graph. Figura extraída de [38].

Technique	AUC (all links)	AUC (hard links)
CN	0.95	0.50
HyperMap	0.96	0.87
DP	0.94	0.59
HyperMap	0.96	0.86
ISP	0.88	0.60
HyperMap	0.96	0.87
Katz	0.96	0.77
HyperMap	0.96	0.87
HRG	0.65	0.53
HyperMap	0.96	0.87

TABLE II
AUC OF CLASSICAL LINK-PREDICTION TECHNIQUES AND COMPARISON TO HYPERMAP.

Under the Receiver Operating Characteristic Curve (AUC) [28]. La misma consiste en asignarle a cada arista no observada entre los nodos i y j un puntaje s_{ij}: cuanto más alto, más probable que exista la arista. El valor de AUC depende de cuantas veces el puntaje de una arista real es mayor que el de una que no existe luego de varias selecciones aleatorias. $AUC = 0{,}5$ es el equivalente a una selección aleatoria y $AUC = 1$ es una selección perfecta.

En [38] primero removieron todos los vértices con grado 1 y 2. Luego removieron al azar el %10, %20 y %30 de las aristas. Las topologías resultantes las embebieron usando HyperMap usando $s_{ij} = p(x_{ij})$ y obtuvieron $AUC = 0{,}963, 0{,}962, 0{,}955$ respectivamente.

Apéndice C

Largo de curvas

C.1. Largo de curvas

En $\mathbb{R}^2$ la noción de distancia está dada por $ds^2 = dx^2 + dy^2$. Esto se interpreta de la siguiente forma: un punto (x, y) que está ligeramente perturbado $(x + dx, y + dy)$ se encuentra a distancia $\sqrt{dx^2 + dy^2}$. ds representa una pequeña variación en la distancia.

Sea una curva en su forma paramétrica $\gamma\colon [a, b] \to \mathbb{R}^2$ tal que $\gamma(t) = (x(t), y(t))$. Usaremos muchos nombres para referirnos a la curva:

$$x = x(t) = Re(\gamma(t))$$
$$y = y(t) = Im(\gamma(t))$$

donde la última igualdad proviene de que la curva vive en los complejos. Por otra parte, la derivada de la curva es $\dot{\gamma}(t) = (\dot{x}(t), \dot{y}(t))$

El largo podemos calcularlo como:

$$\begin{aligned} length_{\mathbb{R}^2}(\gamma) &= \lim_{\Delta t_i \to 0} \sum_i ds_i \\ &= \lim_{\Delta t_i \to 0} \sum_i \sqrt{dx_i^2 + dy_i^2} \\ &= \lim_{\Delta t_i \to 0} \sum_i \Delta t_i \sqrt{\frac{dx_i^2}{\Delta t_i^2} + \frac{dy_i^2}{\Delta t_i^2}} \\ &= \lim_{\Delta t_i \to 0} \sum_i \sqrt{\dot{x} + \dot{y}} \\ &= \int_a^b \sqrt{\dot{x} + \dot{y}} dt \end{aligned} \tag{C.1}$$

Por otra parte, si suponemos una definición distinta de distancia $ds^2 = (dx^2 + dy^2)/y^2$, así como para $\mathbb{R}^2$ también está la del taxista, obtenemos los resultados que se muestran a continuación (la elección de está distancia proviene de la geometría Rimmaniana).

$$\begin{aligned} length_{\mathbb{H}^2}(\gamma) &= \lim_{\Delta t_i \to 0} \sum_i ds_i \\ &= \lim_{\Delta t_i \to 0} \sum_i \sqrt{\frac{dx_i^2 + dy_i^2}{y^2}} \\ &= \lim_{\Delta t_i \to 0} \sum_i \Delta t_i \sqrt{\frac{\frac{dx_i^2}{\Delta t_i^2} + \frac{dy_i^2}{\Delta t_i^2}}{y^2}} \\ &= \lim_{\Delta t_i \to 0} \sum_i \sqrt{\frac{\dot{x} + \dot{y}}{y^2}} \\ &= \int_a^b \frac{\sqrt{\dot{x} + \dot{y}}}{y} dt \\ &= \int_a^b \frac{|\gamma(\dot{t})|}{Im(\gamma(t))} dt \end{aligned} \tag{C.2}$$

Bibliografía

[1] Dimitris Achlioptas y col. «On the bias of traceroute sampling: or, power-law degree distributions in regular graphs». En: *Journal of the ACM (JACM)* 56.4 (2009), pág. 21.

[2] Bernhard Ager y col. «Anatomy of a large European IXP». En: *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM. 2012, págs. 163-174.

[3] J. Ignacio Alvarez-Hamelin y col. «k-core decomposition: a tool for the visualization of large scale networks». En: *CoRR* abs/cs/0504107 (2005). URL: `http://arxiv.org/abs/cs/0504107`.

[4] *Archipielago*. URL: `https://www.caida.org/tools/measurement/skitter` (visitado 27-03-2017).

[5] Brice Augustin, Balachander Krishnamurthy y Walter Willinger. «IXPs: mapped?» En: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM. 2009, págs. 336-349.

[6] Brice Augustin y col. «Avoiding traceroute anomalies with Paris traceroute». En: *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM. 2006, págs. 153-158.

[7] Albert-László Barabási y Réka Albert. «Emergence of scaling in random networks». En: *science* 286.5439 (1999), págs. 509-512.

[8] Robert Braden. *RFC-1122: Requirements for internet hosts*. Inf. téc. 1989, págs. 356-363.

[9] Luca Dall'Asta y col. «Exploring networks with traceroute-like probes: Theory and simulations». En: *Theoretical Computer Science* 355.1 (2006), págs. 6-24.

[10] NetworkX Developers. «NetworkX». En: *networkx. lanl. gov* (2010).

[11] Amogh Dhamdhere y Constantine Dovrolis. «Twelve years in the evolution of the internet ecosystem». En: *IEEE/ACM Transactions on Networking (ToN)* 19.5 (2011), págs. 1420-1433.

[12] Giuseppe Di Battista, Maurizio Patrignani y Maurizio Pizzonia. «Computing the types of the relationships between autonomous systems». En: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 1. IEEE. 2003, págs. 156-165.

[13] Benoit Donnet y Timur Friedman. «Internet topology discovery: a survey». En: *IEEE Communications Surveys & Tutorials* 9.4 (2007), págs. 56-69.

[14] M Faloutsos, P Faloutsos y C Faloutsos. «ACM SIGCOMM'99». En: *Comput. Commun. Rev* 29.251 (1999), pág. 16.

[15] Pawel Foremski, David Plonka y Arthur Berger. «Entropy/ip: Uncovering structure in ipv6 addresses». En: *Proceedings of the 2016 ACM on Internet Measurement Conference*. ACM. 2016, págs. 167-181.

[16] Lixin Gao. «On Inferring Autonomous System Relationships in the Internet». En: *IEEE/ACM Trans. Netw.* 9.6 (dic. de 2001), págs. 733-745. ISSN: 1063-6692. DOI: 10.1109/90.974527. URL: http://dx.doi.org/10.1109/90.974527.

[17] Oliver Gasser y col. «Scanning the IPv6 internet: towards a comprehensive hitlist». En: *arXiv preprint arXiv:1607.05179* (2016).

[18] Mehmet H Gunes y Kamil Sarac. «Analytical IP alias resolution». En: *Communications, 2006. ICC'06. IEEE International Conference on*. Vol. 1. IEEE. 2006, págs. 459-464.

[19] Mehmet H Gunes y Kamil Sarac. «Resolving IP aliases in building traceroute-based Internet maps». En: *IEEE/ACM Transactions on Networking (ToN)* 17.6 (2009), págs. 1738-1751.

[20] Mehmet Hadi Gunes y Kamil Sarac. «Resolving anonymous routers in internet topology measurement studies». En: *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE. 2008, págs. 1076-1084.

[21] B. Huffaker, M. Fomenkov y k. claffy. *Internet Topology Data Comparison*. Inf. téc. Cooperative Association for Internet Data Analysis (CAIDA), 2012.

[22] *Hyperbolic Geometry*. URL: http://www.maths.manchester.ac.uk/~cwalkden/hyperbolic-geometry/hyperbolic_geometry.pdf.

[23] *IPv4 Routed Topology AS*. URL: http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml.

[24] Ken Keys. «Internet-scale IP alias resolution techniques». En: *ACM SIGCOMM Computer Communication Review* 40.1 (2010), págs. 50-55.

[25] Mathieu Lacage. «Outils d'Expérimentation pour la Recherche en Réseaux». Tesis doct. Nice, 2010.

[26] Anukool Lakhina y col. «Sampling biases in IP topology measurements». En: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 1. IEEE. 2003, págs. 332-341.

[27] *LaNet-Vi*. URL: http://lanet-vi.fi.uba.ar/index.php.

[28] Linyuan Lü y Tao Zhou. «Link prediction in complex networks: A survey». En: *Physica A: statistical mechanics and its applications* 390.6 (2011), págs. 1150-1170.

[29] Matthew Luckie. «Scamper: a scalable and extensible packet prober for active measurement of the internet». En: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM. 2010, págs. 239-245.

[30] P. Mahadevan y col. «The Internet AS-Level Topology: Three Data Sources and One Definitive Metric». En: *ACM SIGCOMM Computer Communication Review (CCR)* 36 No 1 (2006), págs. 17-26.

[31] LLC MaxMind. *GeoIP*. 2006.

[32] Alberto Medina y col. «BRITE: An approach to universal topology generation». En: *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*. IEEE. 2001, págs. 346-353.

[33] Dirk Merkel. «Docker: lightweight linux containers for consistent development and deployment». En: *Linux Journal* 2014.239 (2014), pág. 2.

[34] *Merkle Patrica Trie*. URL: https://github.com/ethereum/wiki/wiki/Patricia-Tree.

[35] Reza Motamedi, Reza Rejaie y Walter Willinger. «A survey of techniques for Internet topology discovery». En: *IEEE Communications Surveys & Tutorials* 17.2 (2015), págs. 1044-1065.

[36] Mark EJ Newman. «Random graphs as models of networks». En: *arXiv preprint cond-mat/0202208* (2002).

[37] *NS3*. URL: `https://www.nsnam.org/`.

[38] Fragkiskos Papadopoulos, Constantinos Psomas y Dmitri Krioukov. «Network mapping by replaying hyperbolic growth». En: *IEEE/ACM Transactions on Networking (TON)* 23.1 (2015), págs. 198-211.

[39] Fragkiskos Papadopoulos, Constantinos Psomas y Dmitri Krioukov. «Replaying the geometric growth of complex networks and application to the AS internet». En: *ACM SIGMETRICS Performance Evaluation Review* 40.3 (2012), págs. 104-106.

[40] Fragkiskos Papadopoulos y col. «Popularity versus similarity in growing networks». En: *Nature* 489.7417 (2012), págs. 537-540.

[41] Romualdo Pastor-Satorras y Alessandro Vespignani. *Evolution and structure of the Internet: A statistical physics approach*. Cambridge University Press, 2007.

[42] Maurizio Pizzonia y Massimo Rimondini. «Netkit: easy emulation of complex networks on inexpensive hardware». En: *Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*. ICST (Institute for Computer Sciences, Social-Informatics y Telecommunications Engineering). 2008, pág. 7.

[43] *ripe*. URL: `https://atlas.ripe.net/` (visitado 27-03-2017).

[44] *RIPE RIS*. 10 de abr. de 2017. URL: `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`.

[45] Oregon RouteViews. «University of Oregon RouteViews project». En: *Eugene, OR.[Online]. Available: http://www. routeviews. org* ().

[46] Mario A Sánchez y col. «Dasu: Pushing Experiments to the Internet's Edge.» En: *NSDI*. 2013, págs. 487-499.

[47] Yuval Shavitt y Tomer Tankel. «Hyperbolic Embedding of Internet Graph for Distance Estimation and Overlay Construction». En: *IEEE/ACM Trans. Netw.* 16.1 (feb. de 2008), págs. 25-36. ISSN: 1063-6692. DOI: 10.1109/TNET.2007.899021. URL: http://dx.doi.org/10.1109/TNET.2007.899021.

[48] Yuval Shavitt y Tomer Tankel. «On the curvature of the Internet and its usage for overlay construction and distance estimation». En: *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. Vol. 1. IEEE. 2004.

[49] *Skitter*. URL: https://www.caida.org/tools/measurement/skitter/ (visitado 27-03-2017).

[50] Neil Spring y col. *How to resolve IP aliases*. Inf. téc. Technical report, Univ. of Washington, 2004.

[51] Hongsuda Tangmunarunkit y col. «The impact of routing policy on internet paths». En: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 2. IEEE. 2001, págs. 736-742.

[52] *Team Cymru*. URL: http://www.team-cymru.org/index.html.

[53] *The CAIDA AS Relationship Database*. 10 de abr. de 2017. URL: http://www.caida.org/data/as-relationships/.

[54] Paris Traceroute. «Paris Traceroute». En: *URL: http://www. paris-traceroute. net* (2016).

[55] Bernard M Waxman. «Routing of multipoint connections». En: *IEEE journal on selected areas in communications* 6.9 (1988), págs. 1617-1622.

[56] R Yakov, L Tony y SA Hares. *Border Gateway Protocol4 (BGP-4)*. Inf. téc. 2006.