

Tomografía de Internet: Adquisición y Estudio de Parámetros Dinámicos de la Red

Mauricio Anderson Ricci^{†1}

[†]Facultad de Ingeniería, Universidad de Buenos Aires
 Av. Paseo Colón 850 - C1063ACV - Buenos Aires - Argentina
¹anderson@cnet.fi.uba.ar

Recibido: 04/07/2017; Aceptado: 22/07/2017

Abstract—This work presents an Internet's Topology study, based on its tomography. Our objective was on the gathering of data and their analysis, at both levels: routers and IP. To do that, we developed an Open Source tool called *Magallanes*, which uses PlanetLab platform for performing explorations through nodes around the world.

Resumen— En este trabajo hemos realizado un estudio de la topología de Internet, mediante lo que se conoce en la bibliografía como *Tomografía de Red*. Nuestro objetivo estuvo focalizado en la adquisición y análisis de datos que nos permitiesen caracterizar a Internet, tanto a nivel IP como a nivel de *router*. Para esto hemos desarrollado *Magallanes*, un software de medición que opera sobre la plataforma PlanetLab y nos permite realizar exploraciones desde monitores distribuidos en todo el planeta.

I. INTRODUCCIÓN

En esta sección veremos los conceptos fundamentales sobre lo que se basa este trabajo. Comenzaremos explicando brevemente como funciona Internet para luego hablar sobre lo que se entiende por *tomografía de red*.

A. Funcionamiento de Internet

Los *routers* son elementos que operan en la capa de red. Se encargan de dirigir los paquetes de datos desde el origen hasta el destino entre las diversas redes que hay entre ambos. La decisión de que ruta utilizar para enviar los paquetes es tomada por el protocolo de enrutamiento configurado en cada *router*.

Para ejemplificar la idea de rutas, hemos utilizado la herramienta *traceroute*, la cual permite obtener información del trayecto que siguen los paquetes entre dos ubicaciones, desde un nodo de PlanetLab ubicado en Estados Unidos hasta otro en Nueva Zelanda. Posteriormente, con los resultados del *traceroute* y la ayuda de una herramienta de geolocalización IP hemos generado la Figura 1, donde se observa el camino seguido por el paquete a través del planeta. En cada uno de los puntos donde se observa un cambio de dirección lo que se encuentra es un *router* que tomó la decisión de enviar el paquete por uno u otro camino dentro de los disponibles.

La importancia de estudiar las características de estos caminos se debe a que sobre Internet funcionan un gran número de servicios con alcance global. Estos servicios son los hacen que esta sea indispensable para la sociedad actual. Un ejemplo de estos servicios es la *World Wide Web*, el cual es un servicio de distribución de hipertexto que utiliza Internet como medio de transmisión.

B. Tomografía de la red

Internet, al estar constituida por muchísimas redes, presenta continuas variaciones en su topología. Estas varia-

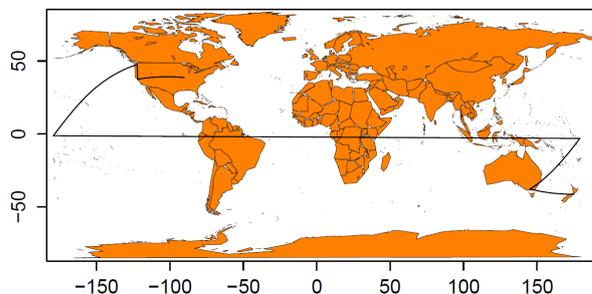


Fig. 1: Ruta seguida por un paquete desde un nodo en EEUU hasta otro en Nueva Zelanda

ciones son causadas principalmente por dos motivos: el primero es que el protocolo TCP/IP es tolerante a fallas, lo cual implica que ante la presencia de fallas el protocolo sigue siendo capaz de entregar los paquetes de datos al destinatario. La otra causa es que al ser operada de forma autónoma por distintas empresas, las mismas pueden introducir cambios en su propia red sin justificar tales cambios ante ninguna entidad.

Por lo ya mencionado, es importantes poder monitorear el estado de la red y así conocer y modelar sus características. Sin esta información resulta imposible desarrollar nuevas tecnologías, protocolos y aplicaciones dado que no se podría conocer ni prever el comportamiento de estos desarrollos al momento de funcionar en entornos reales sobre Internet.

Es acá donde entra en juego el concepto de *tomografía de Red*, o en nuestro caso, *de Internet*. Este termino hace referencia al estudio de las características de la red mediante mediciones externas. Estas reciben el nombre de *externas* por ser realizadas siempre en nodos de la red, sin estudiar lo que sucede con los paquetes mientras viajan. Para ejemplificar, en el caso de la Figura 1 tenemos un paquete que viaja desde un nodo en Estados Unidos hasta otro en Nueva Zelanda a través de distintos enlaces en Internet. Nosotros medimos las características de tales enlaces estando ubicados en uno de los extremos del trayecto, sin necesidad de acceder a cada uno de los enlaces.

C. Etapas de adquisición de datos

Existen dos pasos esenciales en todos los trabajos del tema: la *etapa de exploración*, en la que se adquieren datos de los enlaces, las rutas, el tráfico, etc; y el proceso de *resolución de alias*, que consiste en el mapeo de direcciones IP previamente halladas en la etapa de exploración a *routers*.

Dentro de la etapa de exploración, las técnicas utilizadas se basan principalmente en los conocidos *ping* y *traceroute*, y sus variantes. Estas herramientas nos permiten conocer

diferentes características de las rutas que siguen los paquetes a través de la red.

Por otra parte, los *routers* poseen diferentes interfaces, cada una de las cuales generalmente está conectada a una red distinta y posee su propia dirección IP. Al realizar una exploración mediante *traceroute*, gran parte de las rutas que se obtengan atravesarán *routers* comunes a través de diferentes interfaces. Por lo tanto, es de interés conocer cuales direcciones IP halladas pertenecen a cada uno de estos *routers* con el fin de generar un mapa de la topología de un nivel de jerarquía mayor. Esta tarea de detectar cuales IP pertenecen a un mismo *router* se conoce en la bibliografía como *resoluciones de alises*.

En la resoluciones de alises se hace una primera distinción entre técnicas activas y pasivas (o analíticas) [1]. Las primeras realizan la asociación de IP a *router* enviando sondas (paquetes) a las direcciones a resolver y analizando las respuestas de las mismas. Ejemplo de estas técnicas son Mercator [2], Ally [3] y MIDAR [4]. Luego, las técnicas pasivas realizan la inferencia sin la necesidad de enviar paquetes a la direcciones en cuestión, en cambio utilizan únicamente la información adquirida en la etapa de exploración para inferir los alias. Ejemplos de estos métodos son DisCarte [5] y Kapar [6].

II. ANTECEDENTE Y PROYECTOS DE INVESTIGACIÓN ACTUALES

En primer lugar debemos mencionar a CAIDA [7], institución cuya finalidad es generar y recolectar datos de Internet para su estudio. El primer proyecto desarrollado en CAIDA fue denominado Skitter [8], en el cual se desarrolló una herramienta capaz de recolectar datos de parámetro de Internet. Esta herramienta basaba su operación en el comando *traceroute* [9] que, como resultado de su ejecución, devuelve parámetros de la red asociados a una ruta. Luego, a través de las rutas se infieren los parámetros con los cuales se define el comportamiento de Internet.

Con Skitter se recolectaron datos para un gran número de trabajos de investigación entre los cuales destaca *Internet tomography* [10]. En este trabajo se sentaron las bases de lo que denominamos *tomografía de Internet*. Los autores explican que el termino *tomografía* es útil ya que, al igual que en la práctica médica, se obtiene información del objeto bajo estudio de forma poco invasiva. A su vez, se sentaron cuatro ejes fundamentales al momento de hacer un relevamiento de las características de Internet. Estos son:

- 1) Determinar como se interconectan las redes
- 2) Determinar cuánto demora y por cuántos saltos pasa un paquete entre origen y destino
- 3) Analizar la frecuencia y el patrón que siguen el cambio de las rutas
- 4) Visualizar a través de un grafo las forma que toma Internet

Luego del éxito de Skitter, CAIDA comenzó un nuevo proyecto denominado Archipiélago [11] (abreviado como Ark). Este es un proyecto tiene continuidad hoy en día y busca relevar datos de Internet a través de la herramienta *paris-traceroute* [12], generando *traceroutes* desde varios puntos de medición alrededor del planeta.

DIMES [13], al igual que los proyectos de CAIDA, surge con el objetivo de determinar métricas para caracterizar Internet tales como la capacidad de enlaces, distancias medidas en cantidad de saltos o latencias. Sin embargo la arquitectura de DIMES varia sustancialmente de la implementada en Ark. DIMES se basa en lanzar *traceroutes* en paralelo desde computadoras de usuarios voluntarios que han instalado el software *netDIMES*. Este *software* habilita el uso de la computadora dentro de la plataforma DIMES. Está diseñado de tal forma de servir al proyecto pero al mismo tiempo garantizarle al voluntario la seguridad de su equipo.

iPlane [14] es un proyecto desarrollado por la Universidad de Washington, con cerca de 250 puntos de medición. Busca generar de mapas de Internet con el objetivo de predecir el rendimiento de punta a punta entre dos nodos cualesquiera, partiendo de ciertos enlaces cuyo rendimiento es previamente conocido. Típicamente los enlaces conocidos se tratan de enlaces altamente transitados ubicados en la *backbone* de Internet. El objetivo principal de este proyecto es proveer información sobre la calidad de los enlaces minimizando el número de mediciones requeridas.

El último proyecto es Route Views [15]. Este es coordinado por la universidad de Oregon y en la actualidad recolecta y publica diariamente las tablas de ruteo BGP de cientos de *routers* en todo el mundo. Esta tablas son de gran utilidad para múltiples proyectos de investigación de Internet ya que a partir de las mismas se puede deducir gran cantidad de información. Las tablas BGP cuentan con un campo denominado *AS-path* que indica la secuencia de ASN (Numero de sistema autónomo) para alcanzar la red destino. Es decir, a través de las tablas de ruteo se pueden inferir la topología de los sistemas autónomos.

III. HERRAMIENTA DE EXPLORACIÓN: MAGALLANES

Magallanes [16] es la herramienta de medición que hemos desarrollado para realizar nuestras mediciones y adquirir datos de la estructura de la red. En esta sección veremos cuál fue la motivación que nos llevó a su desarrollo, el entorno de trabajo sobre el que funciona y los módulos que lo componen. Por último, cerramos la sección comentando sobre algunos algoritmos que hemos implementado para lidiar con problemas típicos en mediciones de Internet.

A. Motivación

Debido a la gran cantidad de problemas a afrontar al momento de obtener datos de Internet, la comunidad científica no cuenta con una amplia variedad de herramientas disponibles. En cambio, cuenta con un abanico de herramientas para cada tarea individual, pero que no pueden trabajar por si mismas en conjunto. Esto hace necesario una plataforma con la que se puedan integrar diferentes técnicas, una especie de *Todo en uno*, que simplifique el trabajo de adquisición de datos para el investigador. Por lo tanto, nuestro objetivo es proveer a la comunidad científica de una herramienta flexible que permita la realización de experimentos configurables.

En síntesis, la filosofía que hemos seguido al momento del diseño de Magallanes se resume en lo siguiente:

- 1) **Todo en uno:** desde una única plataforma se puede gestionar todos los pasos necesarios para la adquisición

de la información deseada, esto es, agregar nodos, instalación de software requerido en los mismos, generar las exploraciones, resolución de alias, etc. Todo esto sin necesidad de recurrir a programas externos.

- 2) **Modular:** se pueden agregar nuevas funcionalidades sin realizar grandes cambios en el código principal. Cualquier persona puede hacer un *fork* del repositorio donde se aloja el programa y realizar las modificaciones que se ajusten a sus necesidades.
- 3) **Configurable:** cada exploración se puede configurar según los fines de cada exploración. Dentro de los parámetros configurables estarán, duración de la exploración, cantidad de nodos origen y destino, tipo de *traceroute*, periodo de *traceroute*, etc.

B. Entorno de trabajo

Dada la filosofía del grupo donde se realizó el trabajo, la única restricción que se tomamos al momento de diseñar *Magallanes* fue que se implementase utilizando tecnología *open source*.

Linux: es un sistema operativo de código abierto ampliamente utilizado en ámbitos académicos y de investigación. Por lo tanto, el software fue diseñado para correr sobre este sistema operativo, y en particular bajo distribuciones basadas en *Debian*.

Python: es un lenguaje de programación interpretado de alto nivel, multiparadigma y de código abierto. Lo hemos seleccionado como lenguaje de programación por múltiples motivos. Primero, se trata de un lenguaje con una amplia comunidad de desarrolladores, lo que implica que existan desarrolladas bibliotecas para multitud de tareas, entre las cuales, las necesarias para nuestros propósitos. Entre estos podemos mencionar la posibilidad de interactuar con el sistema operativo y con la base de datos que utilizaremos. A su vez, en el grupo donde se ha llevado a cabo este trabajo se cuenta con una amplia experiencia en su utilización.

Postgres: ha sido el motor de base de datos elegido por contar con los requerimientos esenciales de ser *open source*, funcionar en entornos *Linux*, contar una API para ser operado desde *Python* y ser de tipo relacional. Adicionalmente, cuenta con características extras que lo hacen ideal para nuestro propósito como poseer tipos de datos nativos orientado al uso en redes (direcciones IP y bloques de direcciones con formato CIDR), y funciones para trabajar con las mismas.

PlanetLab: es una plataforma distribuida geográficamente para el desarrollo, evaluación y acceso a servicios de red a escala planetaria [17]. Actualmente la misma es mantenida por un grupo de investigadores en 300 sitios pertenecientes a más de 30 países, y posee 1.353 nodos en 717 sitios.

En la parte operativa, a cada uno de los servidores que posee los denominaremos *nodos*. Los usuarios obtienen acceso a los recursos de un nodo en forma de espacios virtuales, los cuales son denominados *slices*. Los nodos son asociados a un *slice*, y sobre estos se pueden instalar las aplicaciones necesarias para correr diferentes experimentos.

GitHub: es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones *GIT*. Permite alojar en un repositorio archivos de código y además brinda herramientas muy útiles

para el trabajo en equipo, como la posibilidad de hacer un *fork* y solicitar *pull request*. Realizar un *fork* consiste clonar un repositorio ajeno para eliminar algún *bug* o modificar cosas de él. Una vez realizadas las modificaciones se envía un *pull request* al dueño del proyecto solicitando copiar las modificaciones al código original. Éste podrá analizar los cambios que se han realizado, y si considera interesante las contribuciones, aceptarlas.

Scamper: es una herramienta desarrollada por CAIDA para la realización de diferentes tipos de mediciones en Internet. Dentro de estos tipos nos centraremos la implementación de *traceroute* por tener un rol central en la funcionalidad de *Magallanes*.

La versión de *traceroute* que se implementa se denomina *Paris-Traceroute*. Su principal aporte es que permite lidiar con un fenómeno presente en Internet conocido como *balanceo de carga*, el cual, utilizando la versión clásica de *traceroute* haría que se inferan topologías erróneas.

MIDAR: es un algoritmo de resolución de alias desarrollado por CAIDA que implementa varias características que logran hacer que se obtengan resultados más precisos respecto de otros algoritmos. También tiene la posibilidad de ser utilizado con conjuntos de direcciones IP de gran tamaño. Basa su funcionamiento en sondear las direcciones IP a resolver y analizar el campo IP-ID de los paquetes de respuesta generadas por las interfaces de los *routers*.

C. Módulos

El diseño de *Magallanes* lo hemos dividido en dos módulos.

Por un lado, el *modulo administración* se encarga de la gestión del *slice* y demás tareas de mantenimiento. Las acciones que se realizan desde acá son:

- Consultar el estado de los nodos
- Agregar nodos al *slice*
- Quitar nodos del *slice*
- Preparación de los nodos para correr exploraciones
- Mantenimiento de la base de datos

Mientras que por otro lado tenemos el *modulo exploración*. Este se encarga de:

- Programar nuevas exploraciones
- Cancelar exploraciones en ejecución
- Almacenar exploraciones
- Realizar resolución de alias
- Eliminar datos de la base de datos

D. Algoritmos

En esta sección comentaremos algunos de los algoritmos más relevantes que hemos implementado en *Magallanes*.

Selección de target

Al momento de seleccionar las direcciones IP destinos hacia las cuales se realizarán los *traceroutes* hemos optado por incluir cuatro opciones.

Las dos primeras opciones corresponder a usar como destino nodos de *PlanetLab*, seleccionados específica o aleatoriamente. De este modo se permite ejecutar *traceroutes* desde varios extremos entre si mismos. La tercera opción permite indicar manualmente direcciones IP públicas a utilizar como destino.

La última opción corresponde a tomar direcciones IP publicas aleatorias. Como la idea es que una exploración abarque los más posible hemos decidido tratar de utilizar direcciones distribuidas alrededor del planeta. Para esto hemos empleado la base de datos libres GeoLite2 [18] provista por la empresa MaxMind [19]. Esta base de datos cubre todo el espacio de direcciones IPv4, incluyendo todas las direcciones públicas disponibles.

Cálculo de *First-Hop*

Un problema que se genera al realizar *traceroutes* desde un monitor a múltiples destinos es que los primeros *routers* para las distintas rutas serán siempre los mismos, y por lo tanto a estos *routers* se los exigiría mucho y podrían dejar de responder. A su vez, esto hace que no se obtengan mediciones balanceadas de los enlaces, es decir, no se obtendrían la misma cantidad de mediciones en todos los enlaces de la topología bajo estudio.

Para mitigar este problema hemos implementado el Algoritmo 1 que se encarga de hacer que, en una ronda de *traceroutes*, se muestree cada dirección IP descubierta en la topología una única vez. Para lograr esto, el algoritmo realiza una primer ronda de *traceroutes* en la cual cada *traceroute* es ejecutado con el parámetro *tll-inicial* igual 1, es decir, se comienza a muestrear desde el primer *hop*. Una vez finalizada esta ronda inicial, se analizan los resultados y se averigua para cada dirección IP destino cual es el primer *hop* que no es alcanzado en ningún otro *traceroute* hacia las restantes direcciones IP. Finalmente, se lanzan en los siguientes *traceroute* configurando el parámetro *tll-inicial* de modo que el primer paquete de cada *traceroute* alcance a estos *hops*.

Algorithm 1 Calculo de *first hop*

```

Input: target: lista de IP a las cuales se harán traceroutes
Output: first_hop: vector con el TTL inicial a cada IP del target
1:  $IP_{found} = \phi$ 
2:  $first\_hop = \phi$ 
3: for  $i; i++;$   $|target|$  do
4:   traceroute( $IP = IP[i], TTL_{initial} = 1$ )
5: for  $i; i++;$   $|target|$  do
6:   trace = read_traceroute( $IP[i]$ )
7:    $TTL_{initial} = 1$ 
8:   for hop in trace do
9:     if hop in  $IP_{found}$  then
10:       $TTL_{initial}++$ 
11:     else
12:        $IP_{found}$  append hop
13:        $first\_hop[i] = TTL_{initial}$ 
14:     break
15: return first_hop

```

Para comprobar los resultados de la implementación de este algoritmo hemos realizado un experimento que consistió en tomar un monitor desde el cual se realizaron dos exploraciones. En ambas se realizaron 50 rondas de *traceroutes* hacia las mismas 100 direcciones IP, es decir, se realizaron 5.000 *traceroutes*. En una primera exploración se dejó el

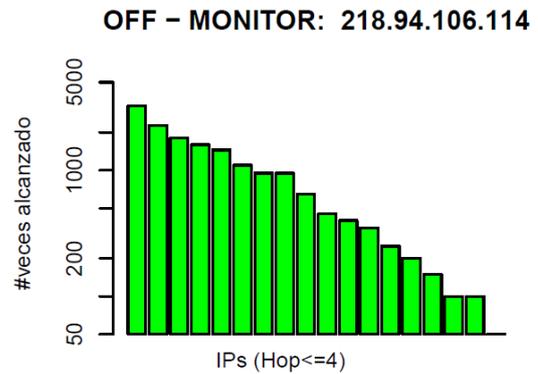


Fig. 2: Algoritmo de *First-Hop* desactivado

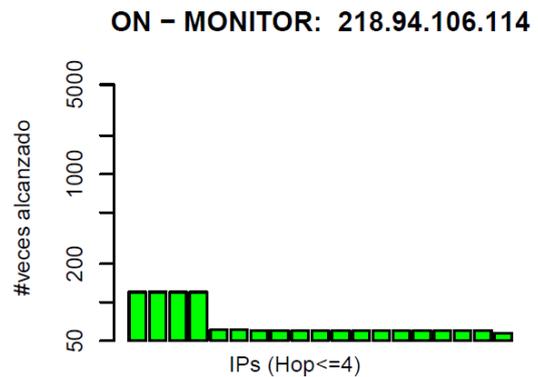


Fig. 3: Algoritmo de *First-Hop* activado

algoritmo desactivado, y luego en la siguiente exploración se lo activo. De cada una de estas exploraciones se tomaron los resultados obtenidos considerando los cuatro primeros *hops* hacia cada destino y se generaron las figuras 2 y 3. En estas, cada barra vertical representa una dirección IP hallada, mientras que la altura es la cantidad de veces que la dirección IP es alcanzada. Se observa que en el segundo caso hay una variación mucho menor que en el primer caso en la cantidad de veces que es alcanzada cada dirección IP. Esto es justamente lo que se buscaba con la implementación del algoritmo.

Resolución de alias

Para la resolución de alias hemos utilizado MIDAR. Este algoritmo tiene dos modos de funcionamiento:

- 1) Modo local: utiliza un único nodo para la resolución, y permite resolver hasta 40.000 direcciones IP por ejecución.
- 2) Modo distribuido: utiliza múltiples nodos en simultáneo y permite resolver conjuntos compuesto de millones de direcciones IP.

Dado que nosotros tenemos que resolver conjuntos compuestos por más de 40.000 direcciones IP, el funcionamiento en modo local no nos es útil. Luego, dado que contamos con múltiples nodos que podríamos emplear, lo ideal sería utilizar MIDAR en modo distribuido, pero no hemos podido implementarlo por problemas de compatibilidad de ciertas bibliotecas necesarias con el sistema operativo Fedora 8 presente en los nodos de PlanetLab.

Ante este problema nos hemos puesto en contacto con

Ken Keys (CAIDA), quien es uno de los desarrolladores del algoritmo. La solución que nos propuso consistió en una implementación mixta.

Lo que finalmente hemos implementado el Algoritmo 2. Este divide inteligentemente nuestro conjunto en subconjuntos no mayores a 40.000 direcciones IP que luego son distribuidos en distintos nodos de PlanetLab. En cada uno de estos nodos es ejecutado el algoritmo en modo local, y luego transferido los resultados hacia el servidor donde se ejecuta Magallanes. De esta forma logramos minimizar la cantidad de direcciones IP alias que van a parar a subconjuntos diferentes y por lo tanto se pierden.

Algorithm 2 Resolución de alias implementada en Magallanes

Input: R_{IP} : lista de IPs halladas;
 R_{node} : lista de nodos con Fedora 14

Output: alias

```

1: alias =  $\phi$ 
2: TCP = UDP = ICMP =  $\phi$ 
3: partitioned  $R_{IP}$  in  $SR_{IP}$  subsets /  $|SR_{IP}(i)| \leq 40000 \forall i$ 
4: for  $i = 1, j = 1; i ++; i \leq \max(|SR_{IP}|)$  do
5:   repeat
6:     method( $SR_{IP}(i), R_{node}(j)$ )
7:     j++
8:     if  $j > j_{max}$  then
9:       j = 1
10:  until method  $\neq \phi$ 
11:  for  $n = 1; n ++; n = |SR_{IP}(i)|$  do
12:    select method(n)
13:    if method(n) = tcp then
14:      insert_ordered(method(n), TCP)
15:    else if method(n) = udp then
16:      insert_ordered(method(n), UDP)
17:    else if method(n) = icmp then
18:      insert_ordered(method(n), ICMP)
19:  for each  $l$  in {TCP, UDP, ICMP} do
20:    select  $S_{IP}(i) =$ 
    { $x \in l/1^{st}$  byte identical and  $|SR_{IP}(i)| \leq 40000$ }
21:    j = 1
22:    repeat
23:      alias_res( $S_{IP}(i), R_{node}(j)$ )
24:      j++
25:      if  $j > j_{max}$  then
26:        j = 1
27:    until alias_res  $\neq \phi$ 
28:    alias  $\leftarrow$  alias + alias_res
29: return alias

```

Los criterios que hemos tomado para generar los subconjuntos, a sugerencia de Ken Keys, son:

- 1) Que las direcciones IP pertenecientes a un mismo subconjunto responda al mismo tipo de sonda (UDP, TCP, ICMP). Basado en mediciones de trabajo previos [20], las direcciones IP que resultan ser alias y responden a distintos tipos de sondas representan solo un 2% del total de alias.
- 2) Buscamos que todas las IP que vayan a parar a un mismo subconjunto posean el mismo primer byte de

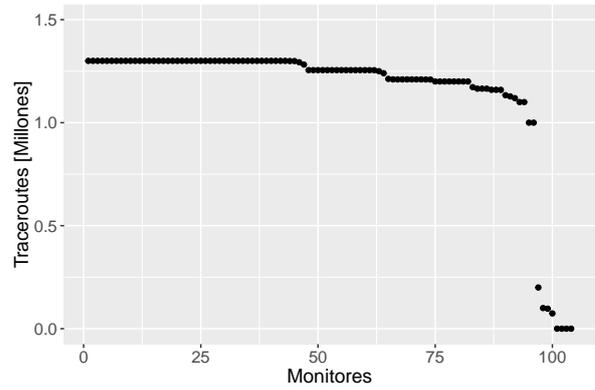


Fig. 4: #Traceroutes realizados en cada monitor

su dirección IP con el objetivo de que las mismas estén geográficamente cercanas entre sí. Esto se debe a que es más probable que dos direcciones IP cercanas geográficamente sean alias, antes que dos direcciones IP distantes.

IV. EXPLORACIÓN DE LA TOPOLOGÍA DE INTERNET

Ahora que hemos descrito el funcionamiento de Magallanes realizaremos una exploración general y analizaremos los resultados obtenidos.

A. Objetivo y preparación

El objetivo que nos hemos propuesto para la exploración es que sea lo más extensa posible a nivel de descubrimiento de *routers* y enlaces, sin focalizar en otros aspectos como la adquisición de datos de latencia o la variación de la topología. Esto es debido a que en toda exploración se debe realizar un balance entre los distintos parámetros según el objetivo que se busque. Finalmente, los parámetros por los que hemos optado son:

- Cantidad de monitores: 104
- Cantidad de destinos: 100.000 direcciones_IP/monitor
- Tiempo entre rondas de *traceroutes*: 3600 seg
- Duración: 12 horas
- Tasa de sondeo (packets-per-second): 350 pps
- Tipo de sonda: UDP-paris
- Recálculo *First-Hop*: Sí (cada 6 horas)
- Resolución de alias: MIDAR

De esta manera esperamos generar idealmente en torno a 125 millones de *traceroutes*.

B. Resultados experimentales y análisis

Una vez descargado los datos de los 104 monitores, lo primero que se observa es que 4 de estos han fallado, mientras que otros 4 han generado una cantidad de *traceroutes* muy inferior a la esperada. En la Figura 4 se muestra la cantidad de *traceroutes* registrados por monitor, ordenados por cantidad. Dado que los monitores que fallaron representan menos de un 10% del total no nos detendremos a analizar las causas de tales fallas.

La cantidad global de direcciones IP descubiertas es 1.318.872. Luego, cruzando los datos con la base de datos GeoLite2 de MaxMind [19], en la Tabla I se muestra como se distribuyen estas a nivel continental.

Continente	#IP	Proporción
Europa (EU)	578.800	43.9%
América del Norte (NA)	378.369	28.7%
Asia (AS)	231.092	17.5%
América del Sur (SA)	50.230	3.8%
Oceanía (OC)	26.726	2.0%
África (AF)	21.577	1.6%
Antártida (An)	18	0.0%
no localizables	32.065	2.4%

TABLE I: Cantidad de direcciones IP halladas por continental

	EU	NA	AS	OC	SA	AF
EU	977.779	148.228	21.722	902	6.629	5.920
NA	148.228	517.006	35.590	5.352	14.193	5.179
AS	21.722	35.590	394.699	4.374	422	1.277
OC	902	5.352	4.374	34.792	12	294
SA	6.629	14.193	422	12	39.077	119
AF	5.920	5.179	1.277	294	119	19.367

TABLE II: Cantidad de enlaces hallados entre continente a nivel IP

Luego, las 1,32 millones de direcciones IP halladas se vinculadas mediante 2,35 millones de enlaces. La distribución de los enlaces a nivel de continente se indica en la Tabla II.

Pasando a la resolución de alias obtenemos que, de las 1,32 millones de direcciones IP se han encontrado, hallamos 102.606 alias agrupando en total a 402.562 direcciones IP (31% del total). Una vez completado este paso ya poseemos los grafos que caracterizan a la red: el grafo a nivel IP y el grafo a nivel *router*.

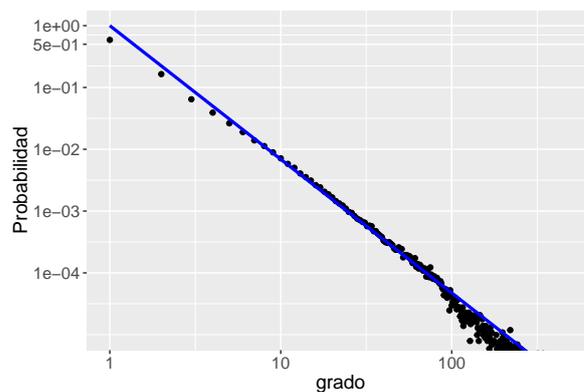
Ahora continuaremos con el análisis del grafo a nivel *router*, describiéndolo mediante las principales métricas utilizadas en análisis de grafos. A este lo consideraremos de ahora en más como *no dirigido*, es decir, no le asignamos sentido al tráfico de paquetes, y no le asociaremos peso a sus enlaces.

Distribución de grado

Dentro de grafos no dirigidos, se define como grado de un nodo a la cantidad de enlaces que este posee. Una caracterización natural de las propiedades estadísticas de los grafos es proporcionada por la distribución de grado de los nodos que lo conforman, lo cual también se puede entender como la probabilidad $P(k)$ de que un nodo aleatorio tenga grado k . De esta manera, la distribución de grado se define según la expresión: $P(k) = \frac{1}{n} \sum_{k_i=k} 1$, donde n es el número de nodos total y k_i el grado del nodo i .

Estudios previos sobre el tema de *redes complejas* han revelado que los grafos que muestran una distribución de probabilidad $P(k)$ con cola pesada en muchos casos pueden aproximarse con precisión por una distribución con comportamiento de *ley de potencias*, esto es $P(K) \sim k^{-\gamma}$, donde $2 \leq \gamma \leq 3$. Esto ha llevado a la introducción del concepto de *red sin escala* [21], las cuales se caracterizan por poseer un pequeño grupo de nodos altamente conectados, mientras que el grado de conexión de los nodos es bastante bajo. Como veremos, Internet presenta este tipo de comportamiento.

En la Figura 5 se muestra la distribución de grado de $P(k)$ en función de k para el grafo obtenido en la exploración y como el mismo efectivamente sigue una ley de potencias. En particular, utilizando la función


 Fig. 5: Ajuste de distribución de grado mediante $P(k) \sim k^{-2.162}$, para $k_{min} = 5$

`power.law.fit{igraph}` de **R**, la cual se basa en el método de Newman [22], se observa que el valor de γ ronda en torno a 2.16 para grado mayor a 5.

Distribución del grado medio de los vecinos

El grado medio de los vecinos de un nodo se define, como su nombre lo indica, como el promedio de grado de los nodos vecinos a este:

$$k_{nn}(k) = \frac{1}{n_k} \sum_{j/k_j=k} \frac{1}{|V(j)|} \sum_{i \in V(j)} k_i$$

dónde $V(j)$ es el conjunto de los vecinos del nodo j , $|V(j)|$ su cardinalidad, k_i es el grado del nodo i , y n_k el número de nodos de grado k .

Este parámetro da una idea de cómo están conectados los nodos de un grafo. En el caso que los nodos de grado elevado posean un k_{nn} pequeño, entonces ellos actuarán de concentradores, implicando que los nodos de grado bajo tengan como vecinos otros de grado elevado. Este comportamiento está definido como *discordante* [23] y se observa en la topología de los sistemas autónomos. En cambio cuando los nodos de grado elevado tienen como vecinos otros nodos de grado también elevado, se denominan *concordante* [23]. Este último comportamiento es observado en algunas redes sociales.

Las nociones de discordante o concordante tienen sentido cuando las pendientes son marcadas, de otro modo las pequeñas variaciones pueden ser originadas por los sesgos en la obtención de los mapas [24]. En la Figura 6 se observa justamente este último caso para el mapa de *routers* que hemos hallado, las variaciones son pequeñas (menos de una década para grado menor que 100). Luego, en la Figura 7 se observa que la mayoría de los nodos están conectados a otros nodos de bajo grado. Esto se explica dada la estructura jerárquica de la red, en la que cada nodo está actuando de puente entre niveles jerárquicos.

Distribución de coeficiente de clustering

El coeficiente de clustering C_i de un nodo mide el nivel de interconexión de los vecinos de este, es decir, mide la probabilidad de que los nodos vecinos a este estén conectados entre sí:

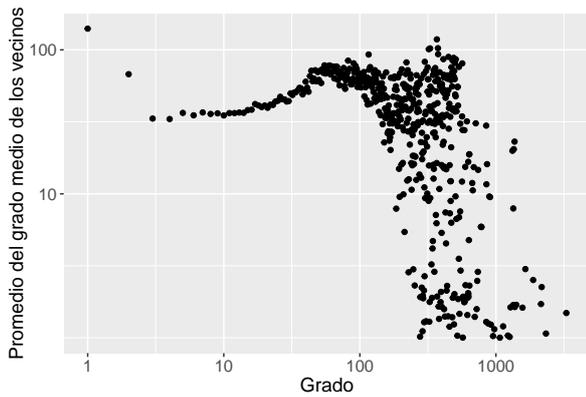


Fig. 6: Promedio del grado medio de los vecinos en función del grado

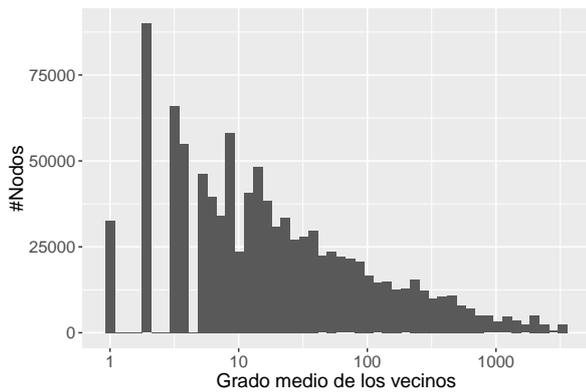


Fig. 7: Histograma del grado medio de los vecinos

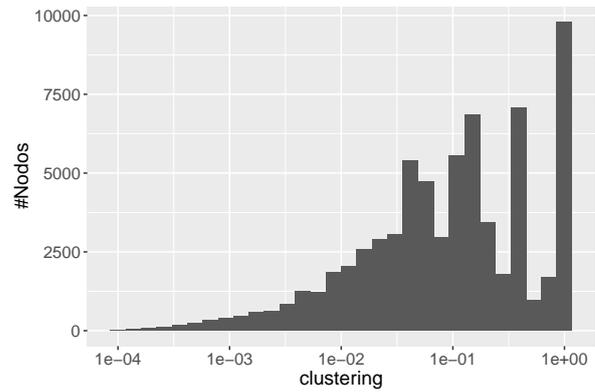


Fig. 8: Histograma del coeficiente de clustering

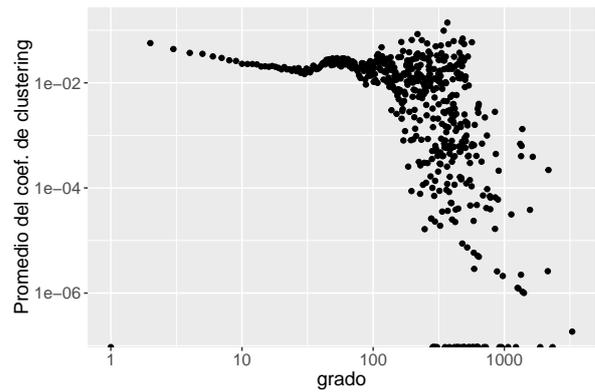


Fig. 9: Promedio del coeficiente de clustering en función del grado

$$C_i = \frac{2 \cdot n_{enlace}}{k_i \cdot (k_i - 1)}$$

donde n_{enlace} es el número de conexiones entre los vecinos del nodo i , y k_i es el grado del nodo i . La función de distribución del grado es:

$$C(k) = \frac{1}{n_k} \cdot \sum_{j/k_j=k} C_j$$

Por ejemplo, si el nodo está agrupado como un *clique* su valor es 1 (máximo), mientras que un valor pequeño indica un nodo poco agrupado en la red. Formalmente, el coeficiente de *clustering* de un nodo está dado por la proporción entre los enlaces conectados con sus vecinos respecto del número de enlaces que habría en un *clique* en el que la conectividad es máxima.

En la Figura 8 se muestra el histograma del coeficiente de clustering del grafo hallado. Se aprecia que en general los nodos tienen un bajo nivel de interconexión en la red, mientras que un grupo tienen un coeficiente cercano a 1. Estos últimos son nodos de bajo grado en los que es más probable encontrar que pertenecen a un *clique*. En cambio, en los nodos de mayor grados, esto no se da en ningún caso dado la enorme cantidad de enlaces que se necesitarían para tal hecho. Estos conceptos se visualiza más claramente en la Figura 9, donde vemos que a medida que aumenta el grado de los nodos disminuye el coeficiente de *clustering*.

Centralidad

La centralidad de un nodo dentro de un grafo se refiere a una medida posible de este en dicho grafo que determina su importancia relativa en la red. Las medidas de centralidad se pueden agrupar en dos categorías: las medidas radiales y mediales. Las primeras toman como punto de referencia un nodo dado que inicia o termina recorridos por la red, mientras que las segundas toman como referencia los recorridos que pasan a través de un nodo dado. Las medidas radiales a su vez se pueden clasificar en medidas de volumen y de longitud, según el tipo de recorridos que consideran. Las primeras miden el volumen (o el número) de recorridos limitados a dicha longitud prefijada, en tanto que las segundas miden la longitud de los recorridos necesarios para alcanzar un volumen prefijado. En base a esta clasificación existen cuatro medidas que son ampliamente utilizadas en análisis de redes:

- La centralidad de grado (*degree centrality*)
- La cercanía (*closeness*)
- La intermediación (*betweenness*)
- La centralidad de vector propio (*eigenvector centrality*)

La primera y la última son medidas radiales de volumen. La segunda es una medida radial de longitud, y la tercera una medida medial.

La centralidad de grado corresponde al número de enlaces que posee un nodo con los demás. Existen criterios que son usados para normalizar esta medida: dividir el grado de

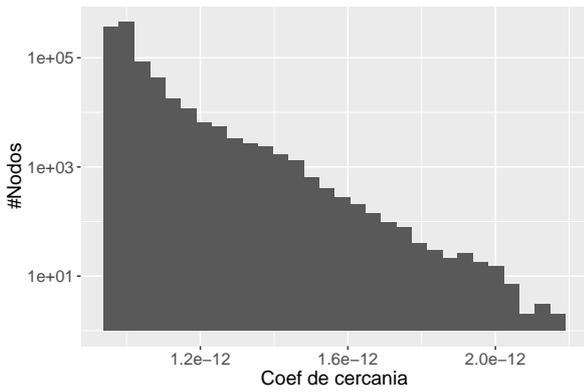


Fig. 10: Histograma de coeficiente de cercanía

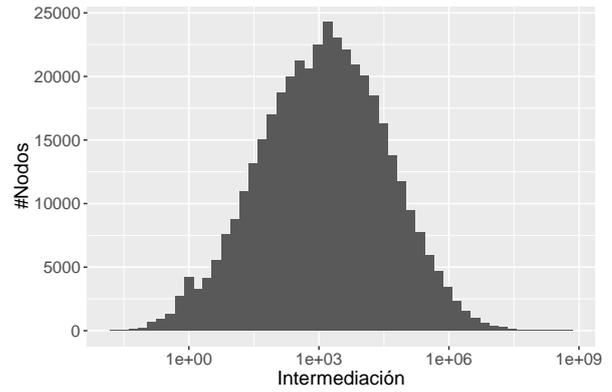


Fig. 12: Histograma de coeficiente de intermediación

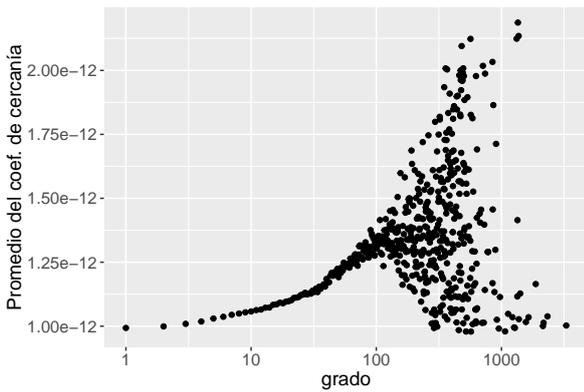


Fig. 11: Promedio del coeficiente de cercanía en función del grado

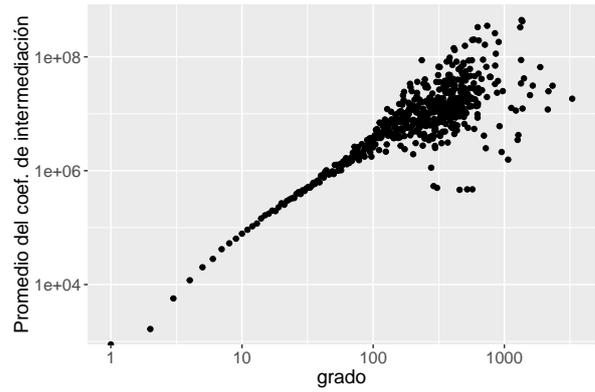


Fig. 13: Promedio del coeficiente de intermediación en función del grado

cada nodo por el máximo grado obtenido de la red, o bien dividirlo por el número total de nodos de la red. Como esta medida es similar entregaría resultados similares a la distribución de grado que vimos previamente no ahondaremos en la misma.

Respecto de la cercanía, dada su definición, se espera que los nodos presentes en el centro de la red (troncal o *backbone*) presenten un nivel de cercanía inferior que los ubicados en las periferias. En la Figura 10 se observa el histograma que presenta este parámetro en nuestra exploración. Como es de esperar, dada la forma jerárquica de la red, se obtiene una mayor densidad de nodos con valor bajo de este parámetro. Luego, en la Figura 11 se observa esta misma característica. A medida que aumenta el grado de los nodos, aumenta el coeficiente de cercanía.

La intermediación de un nodo es una medida que cuantifica la frecuencia o el número de veces que un nodo actúa como un puente a lo largo del camino más corto entre cada par de nodos. Un nodo con un alto grado de intermediación tiene una gran influencia en el tráfico de la red, que en el caso de Internet corresponden a los nodos ubicados en el centro de la red. En la Figura 12 se observa el histograma de este parámetro utilizando una escala logarítmica sobre el eje horizontal. Luego, en la Figura 13 se observa esto mismo. A medida que aumenta el grado de los nodos, mayor tiende a ser su coeficiente de intermediación.

Visualización de topología

Para la visualización del grafo utilizaremos la herramienta *LaNet-vi* [25], la cual se basa en la descomposición en k -núcleos. Para esto introduciremos algunos conceptos necesarios para describir los resultados obtenidos [26]:

- **k -núcleo:** un subgrafo $H = (C, E|C)$ inducido por el conjunto $C \subseteq V$ en un k -núcleo (o un núcleo de orden k) si para todo $v \in C$: $\text{grado}_H(v)$ y H es el máximo subgrafo con esta propiedad. Una forma de obtener la descomposición en k -núcleos es ir eliminando recursivamente todos los nodos de grado menor que k , hasta que todos los nodos restantes tengan grado mayor o igual que k .
- **capa:** un nodo tiene número de capa c (*Shell index*) si dicho nodo pertenece al c -núcleo pero no al $(c + 1)$ -núcleo. Lo importante de este parámetro es su relación con la conectividad. Si bien un k -núcleo no es k -conexo en general, en ciertos casos prácticos se puede dar esta conectividad. Si un grafo es core-conexo, entonces existen k caminos independientes entre todo par de vértices que pertenecen a un k -núcleo. En otras palabras, dado dos vértices u y v , existen al menos $\min(c_u, c_v)$ caminos independientes entre ambos. En lenguaje de redes de datos esto significa que si alguno de esos caminos está obstruido, ya sea por congestión o por una falla momentánea, entonces se puede elegir otro camino para llegar al destino. Incluso, si se

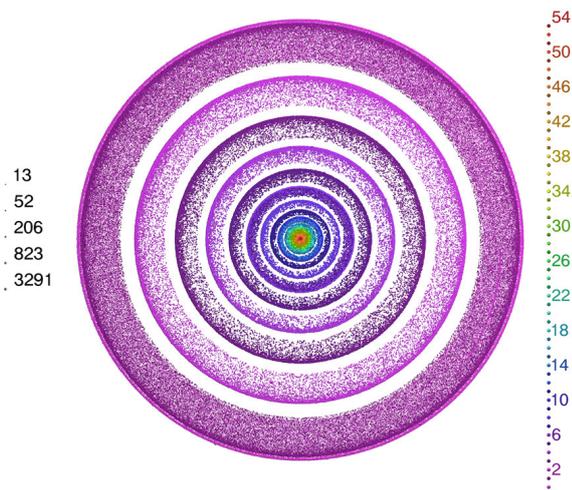


Fig. 14: Visualización del grafo obtenido con Lanet-vi:
Vista completa

necesita una red que cumpla con cierta calidad de servicio, es decir que respete ciertas cotas mínimas para algún parámetro (demora, capacidad, etc.), cuanto más caminos existan entre sus vértices, más posibilidades tendrá de encontrar aquel que verifique la calidad deseada.

Entonces, en la Figura 14 se muestra el grafo obtenido visualizándolo mediante esta técnica. La idea de la visualización es ubicar en el centro la capa con el k_{max} -núcleo en forma de círculo, donde su diámetro es proporcional al número de elementos que lo componen y luego las otras capas k en circunferencias concéntricas que se alejan del k_{max} y donde la circunferencia exterior corresponde a la capa con k_{min} . De esta manera cada circunferencia corresponde a cada capa k y todos los vértices que la conforman se dibujan sobre ellas con el mismo color. A medida que nos acercamos al núcleo se van solapando algunas de estas circunferencias por lo cual puede parecer que poseen más de un color, pero realmente son circunferencias solapadas. Luego, si bien los vértices están ubicados inicialmente en una circunferencia, éstos pueden desplazarse hacia el centro según el valor promedio de c_v de sus vecinos v . Esto se observa más notoriamente en las capas externas.

Sobre la derecha de la imagen se muestra una escala de colores con el número de capa c_i , mientras que a la izquierda una escala logarítmica del grado de los vértices representado por el tamaño de los mismos. Los detalles de la visualización se pueden encontrar extensivamente descritos en [27].

En estas visualizaciones se puede apreciar la propiedad jerárquica de Internet, en la cual cada una de las capas está densamente poblada y las conexiones se producen principalmente entre elementos de capas contiguas, y no hacia la capa central como ocurre en otro tipos de redes tal como la de los sistemas autónomos [24]. Otra observación importante es que no existe una fuerte correlación grado-capo. Esto último se puede ver ya que en las capas exteriores tenemos vértices de gran tamaño o grado elevado.



Fig. 15: Visualización del grafo obtenido con Lanet-vi:
Ampliación sobre la zona central del grafo

V. CONCLUSIONES

A. Contribuciones

Este trabajo ha sido un resumen de mi tesis de grado de Ingeniería Electrónica de la Universidad de Buenos Aires [28]. Como resultado hemos desarrollado una herramienta, Magallanes, para gestión de exploraciones de Internet que cumple con los requisitos que nos propusimos inicialmente. Con esta herramienta hemos realizado distintos estudios.

En primer lugar realizamos una exploración de la estructura de Internet que nos permitió hacer una caracterización a rasgos generales de su topología. A nivel cualitativo los resultados obtenidos son consistentes con lo que esperábamos a priori, demostrando como la red a nivel de router presenta un comportamiento jerárquico, donde los nodos se comunican hacia el centro de la red a través de otros *routes*, y no directamente como sucede en otras redes como la de los sistemas autónomos.

Sumado a esto, se utilizó Magallanes para llevar a cabo trabajos de investigación. Uno sobre la caracterización del parámetro RRT en los primeros *hops* de distintas redes, y luego otro trabajo sobre la influencias de las redes MPLS dentro de la estructura de Internet, que a su vez desembocó en la publicación [29].

B. Trabajos futuros

Para continuar con el desarrollo de este trabajo surgen varias posibilidades de investigación. Por un lado se podría hacer un análisis de las redes por regiones geográficas, es decir, nosotros en el presente trabajo tratamos a la red como un todo, pero podríamos desear caracterizar a la red en cada continente y luego comparar su estructura. A su vez se podría ampliar el análisis de la topología incluyendo peso a los enlaces de modo que representen distintos parámetros como pueden ser latencias, disponibilidad o tasa de pérdida. Otra posibilidad sería integrar una herramienta para mapear el prefijo IP al ANS de cada nodo y luego, mediante la información provista por el proyecto *route views*, realizar el mapeo de cada nodo al sistema autónomo al que pertenece y crear así mapas a nivel de sistema autónomo.

REFERENCES

- [1] K. Keys, "Internet-Scale IP Alias Resolution Techniques," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 40, no. 1, pp. 50–55, Jan 2010.
- [2] R. Govindam and H. Tangmunarunkit, "Heuristics for internet map discovery," 2000.
- [3] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies using rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 2–16, 2004.
- [4] "Midar," <http://www.caida.org/tools/measurement/midar/>.

- [5] R. Sherwood, A. Bender, and N. Spring, "DisCarte: A Disjunctive Internet Cartographer," *ACM SIGCOMM Computer Communication Review*, 2008. [Online]. Available: <http://www.rob-sherwood.net/discarte-sigcomm08.pdf>
- [6] "Kapar," <http://www.caida.org/tools/measurement/kapar/>.
- [7] "Caida:center for applied internet data analysis," <http://www.caida.org/>.
- [8] "Skitter," <http://www.caida.org/tools/measurement/skitter/>.
- [9] V. Jacobson and S. Deering, "traceroute," Feb 1989.
- [10] K. Claffy, T. Monk, and D. McRobb, "Internet Tomography," *Nature*, January 1997. [Online]. Available: <http://www.nature.com/nature/webmatters/tomog/tomog.html>
- [11] "Archipelago (ark) measurement infrastructure," <http://www.caida.org/projects/ark/>.
- [12] "Paris traceroute," <http://www.paris-traceroute.net/>.
- [13] E. S. Yuval Shavitt, "Dimes: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, 2005.
- [14] "iplane," <http://web.eecs.umich.edu/~harshavm/iplane/>.
- [15] "University of oregon route views project," <http://www.routeviews.org/>.
- [16] "Magallanes," <https://github.com/ihameli/magallanes>.
- [17] "Planetlab: An open platform for developing, deploying, accessing planetary-scale services," <https://www.planet-lab.org/>.
- [18] "Geolite2," <http://dev.maxmind.com/geoip/geoip2/geolite2/>.
- [19] "Maxmind," <https://www.maxmind.com/es/home>.
- [20] K. Keys, Y. Hyun, M. Luckie, and k. Claffy, "Internet-Scale IPv4 Alias Resolution with MIDAR," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 383–399, Apr 2013.
- [21] R. Albert, H. Jeong, and A. Barabasi, "Diameter of the world-wide web," *Nature*, Sep 1999.
- [22] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM*, vol. 4, no. 51, pp. 661–703, 2009. [Online]. Available: <http://arxiv.org/pdf/0706.1062.pdf>
- [23] M. E. J. Newman, "Assortative mixing in networks," *Phys. Rev. Lett.*, vol. 89, p. 208701, Oct 2002. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevLett.89.208701>
- [24] J. I. Alvarez-Hamelin, "Taxonomia de los Modelos de Topologia de Internet," in *Mecanica Computacional, Interdisciplinary Mathematical Methods*, vol. XXV, no. 29. CIMEC, 2006, pp. 2597–2612. [Online]. Available: <http://www.cimec.org.ar/ojs/index.php/mc/article/view/636/604>
- [25] "Lanet-vi," <http://lanet-vi.fi.uba.ar/>.
- [26] M. G. Beiró, J. I. Alvarez-Hamelin, and J. R. Busch, "A low complexity visualization tool that helps to perform complex systems analysis," *New J. Phys.*, vol. 10, no. 12, p. 125003, 2008. [Online]. Available: <http://dx.doi.org/10.1088/1367-2630/10/12/125003>
- [27] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani, "Large scale networks fingerprinting and visualization using the k -core decomposition," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, pp. 41–50. [Online]. Available: http://cnet.fi.uba.ar/ignacio.alvarez-hamelin/pdf/AH_D_B_V_NIPS2006.pdf
- [28] M. Anderson, "Tomografía de Internet: Adquisición y estudio de parámetros dinámicos de la red," *FIUBA*, 2016. [Online]. Available: http://cnet.fi.uba.ar/mauricio_anderson/Tesis_Mauricio_Anderson.pdf
- [29] F. D. Revelo, M. A. Ricci, B. Donnet, and J. I. Alvarez-Hamelin, "Unveiling the MPLS Structure on Internet Topology," *8th International Workshop on Traffic Monitoring, Analysis (TMA)*, Apr 2016. [Online]. Available: <http://orbi.ulg.ac.be/bitstream/2268/194694/1/paper.pdf>