



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Miércoles 17 de junio de 2015

Teoría de las Comunicaciones

Integrante	LU	Correo electrónico
Nahuel Lascano	476/11	laski.nahuel@gmail.com
Nicolás Lasso	763/10	lasso.nico@gmail.com
Roberto Rama	490/11	bertoski@gmail.com
Pablo Somodi	818/10	psomodi@dc.uba.ar



Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellon I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Introducción	2
2. Implementación	3
3. Experimentación	4
4. Resultados	6
4.1. Aclaraciones generales	6
4.2. Estimación de RTT y análisis de parámetros	6
4.3. Berkeley, EEUU	8
4.4. Cochin, India	10
4.5. Reykjavik, Iceland	12
4.6. Perm, Rusia	14
4.7. Pretoria, South Africa	16
5. Conclusiones	18

1. Introducción

En el presente trabajo práctico tuvimos como desafío implementar una herramienta muy común de diagnóstico de red, el traceroute. Por medio de la misma los administradores de red pueden analizar la ruta que sigue un paquete hasta un destino dado.

Luego se nos pidió usar la herramienta diseñada para evaluar las rutas que toman los paquetes hacia diferentes lugares del mundo, incluyendo un análisis sobre cómo distinguir los enlaces trasatlánticos y evaluando su correcitud posteriormente con herramientas de geolocalización de IP.

Por último se nos pidió estimar el valor del *throughput* de la conexión para las rutas analizadas, teniendo en cuenta la pérdida de paquetes.

2. Implementación

Los fundamentos del *traceroute* son muy sencillos. Se utilizan paquetes IP para recorrer la ruta aumentando de un salto a la vez. Esto se hace por medio del campo TTL (*Time to live*) que nos permite configurar la cantidad máxima de “saltos” que se le permiten al paquete para llegar a destino. Cuando un router recibe un paquete para redireccionarlo, previo a esto decrementa en 1 su TTL, si el mismo resulta igual a 0 entonces devuelve el mensaje **ICMP Time Exceeded** a la dirección de origen. La herramienta se basa en este comportamiento para obtener la información de la ruta. La misma envía varios paquetes incrementando el TTL progresivamente desde 1 hasta que el paquete efectivamente llega a destino, y va registrando los paquetes de respuesta **ICMP Time Exceeded** que recibe de cada router.

Esta herramienta se puede implementar con paquetes ICMP, TCP o UDP, según necesidad. La herramienta default que se ofrece en cualquier distribución de Linux permite realizar el procedimiento con los tres tipos de paquetes. Para implementar la herramienta utilizamos paquetes de tipo ICMP. Primeramente decidimos resolver el registro DNS a su IP correspondiente para evitar considerar el tiempo que pudiera tardar este procedimiento. Además, esto nos permite evitar el posible cambio de rutas producto de alguna universidad usando Round-robin DNS¹.

En el comienzo de nuestra implementación nuestra herramienta explora la ruta completa para estudiar los hosts que pertenecen a la misma. Decidimos limitar el crecimiento del *timetolive* a 35 ya que es un poco mayor que la que utilizan las herramientas provistas por Linux. De todos modos esto no afectó ya que llegamos a todos los hosts elegidos en menos saltos.

Luego de esto utilizamos la ruta descubierta para realizar 50 repeticiones de mediciones de RTT de forma incremental en el TTL. Utilizamos el promedio de las mismas con el fin de tener valores más estables. Luego de calculado este promedio realizamos las restas uno a uno entre hops sucesivos para obtener los tiempos diferenciales a partir de los acumulados.

Para normalizar los datos para el análisis calculamos el valor Z del RTT de cada hop², que nos da una idea de cuánto se aleja cada RTT de la media, medido en desvíos standard. Para ello usamos la ecuación facilitada por el enunciado:

$$ZRTT_i = \frac{RTT_i - \overline{RTT}}{SRTT}$$

donde \overline{RTT} es el promedio y $SRTT$ el desvío standard de la distribución de los RTTs para una ruta dada. Esto nos permite analizar los saltos con una información estadísticamente más sólida que la diferencia absoluta medida en *ms*.

Para simular el throughput utilizamos la ecuación de Mathis

$$\frac{MSS}{EstimatedRTT * \frac{1}{1 - \sqrt{EstimatedPacketLossProbability}}}$$

donde *EstimatedRTT* se obtiene iterando sucesivas veces la ecuación

$$\alpha * EstimatedRTT + (1 - \alpha) * SampleRTT$$

y *EstimatedPacketLossProbability* se calcula como

$$1 - \frac{\#Echoreply}{\#Echorequest}$$

Para resolver la geolocalización inicialmente decidimos incorporar una parte de código que obtuviera la información para cada IP. Esto nos trajo problemas ya que la diferencia entre los RTT no coincidía con la geolocalización real del salto. La herramienta que nos permitió encontrar estas inconsistencias fue IPLocation³, un meta-buscador que realizaba consultas a la base de IP2Location⁴ actualizada el 01/06/2015.

¹https://en.wikipedia.org/wiki/Round-robin_DNS

²Recordar que nos estamos refiriendo a la diferencia entre el RTT absoluto de un hop y el anterior

³<http://www.iplocation.net/>

⁴<http://www.ip2location.com/>

3. Experimentación

Durante la experimentación encontramos ciertas anomalías detectadas al correr los experimentos. Las mismas fueron:

- Hops que no respondían. En base a este problema se generó la hipótesis de que había hops en los cuales los paquetes ICMP enviados llegaban con un *Time-Exceeded* a un router que estaba configurado para no responder a estos paquetes o para filtrarlos directamente. En base a esto se configuró el método *SR1()* de Scapy para que dado cierto *Timeout* el mismo dejara de esperar a que el router le respondiera permitiéndonos aumentar el TTL y llegar al siguiente Hop.
- Corridas en las cuales no se llegaba a ningún destino haciendo que el script nunca terminara incluso aumentando el TTL y el tiempo de Timeout. Con la misma hipótesis que la anterior, realizando experimentaciones a diferentes universidades, hubo algunas, como las que se muestran a continuación, cuyo servidor estaba configurado para no responder. Por ejemplo:
 - Universidad de Tokio
 - Melbourne University
 - University of Mumbai

En este caso, lo que se realizó para constatar este comportamiento previamente fue hacer un *PING()* a los destinos para corroborar si los mismos responden. De esta manera se supo si había muchos hops que no respondían, como sucedía anteriormente, entre la IP de origen y la de destino o si era efectivamente el destino.

- Las rutas al mismo destino cambiaban. Al correr muchas veces a la misma IP de destino notamos que las rutas podían variar, con lo cual no era lo suficientemente fiable el cálculo del RTT. Dado esto, implementamos una verificación para descartar una corrida cuando se detectaba que una ruta no era igual a la primer ruta obtenida. De esta forma se aseguró la misma ruta en todos los casos. Otra solución posible hubiese sido enviar los paquetes directamente a los hops una vez fijada la ruta, pero preferimos no utilizar esta alternativa ya que la ruta directa hacia cada uno de los hops puede ser muy disinta a la que nos llevaba a ellos cuando nuestro destino era otro (corroborado en pruebas preliminares).
- RTTs no lineales, es decir, si bien se supone que a medida que se realizan saltos a hops más lejanos el RTT debería aumentar, esto no siempre sucedía. Al realizar las experimentaciones se notó que para cierto rango de IPs que pertenecen a la misma región la diferencia entre los RTTs podía ser negativa. Esto puede deberse a que muchos routers encolan las respuestas de ICMP de modo tal que tardan mucho más en responderlas que en reenviarlas. Para confirmarlo o descartarlo, como trabajo futuro se podría implementar el mismo *traceroute* pero con paquetes TCP.
- Otra anomalía ocurría cuando para hops muy distantes la diferencia era muy pequeña. Tras revisar las locaciones de las IPs llegamos a la conclusión de que, si bien hay rangos IPs que pertenecen a la misma región y que deberían estar relativamente cercanos, sucede que la IP física se encuentra realmente en otra locación provocando que se produzca más de un salto transatlántico y que los valores de RTT varíen de esta manera. Estos casos se pueden apreciar en la sección de resultados.

Las universidades a las que finalmente se les realizó el traceroute fueron:

- Universidad de Berkeley, California, EEUU. Con la dirección: berkeley.edu
- Perm State University, Rusia. Con la dirección: en.psu.ru
- Iceland University, Iceland. Con la dirección: english.hi.is
- Cochin University of Science and Technology, Cochin, India. Con la dirección: cusat.ac.in
- University of Pretoria, Pretoria, South Africa. Con la dirección: up.ac.za

Dichas universidades respondieron a los PING previos para realizar la experimentación y todas pertenecen a distintos continentes.

El throughput fue estimado mediante la ecuación de Mathis, descripta en la sección de implementación. Para elegir el valor del α y de la cantidad de iteraciones de la ecuación que calcula el EstimatedRTT (n) realizamos pruebas con $\alpha = 0,1; 0,2; \dots ; 1$ y $n = 10, 20, \dots, 50$.

4. Resultados

4.1. Aclaraciones generales

Para cada ruta elegida realizamos una tabla con los resultados de nuestro *traceroute*, un gráfico con barras representando cada salto y un mapa que marca la ruta geográfica que traza la conexión establecida. En las tablas incluimos el valor promedio del RTT absoluto de la computadora local hasta cada salto, pero como ya fue mencionado calculamos el ZRTT usando la diferencia de RTT entre un salto y el siguiente. En los mapas los saltos dentro de una misma ciudad no aparecen representados.

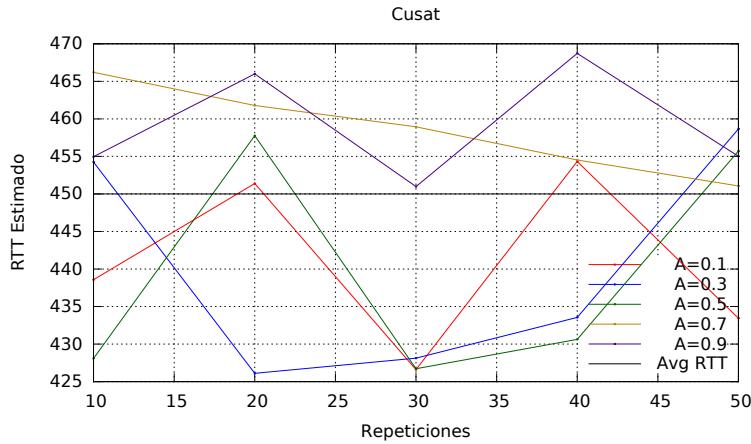
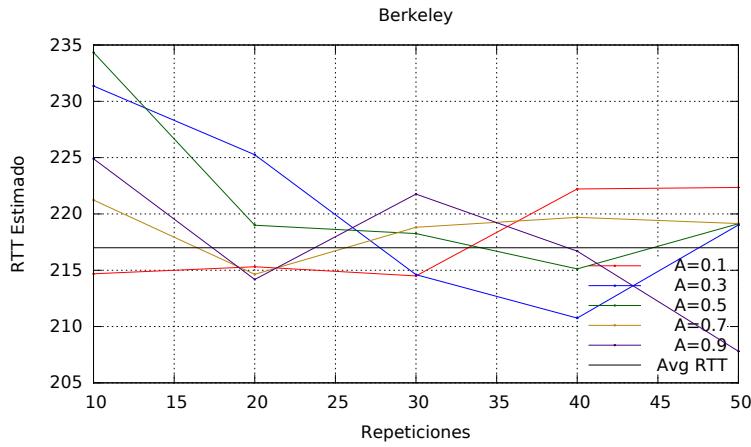
Establecemos como umbral del ZRTT un valor de 1, ya que para el cálculo del mismo se efectúa una normalización, y consideramos que un valor mayor a un desvío estándar comienza a ser estadísticamente relevante. Solo un 15 % de los valores caen por encima del mismo.

Además, agregamos también para contrastar el resultado del enviar 100 paquetes mediante el comando *ping* de Linux.

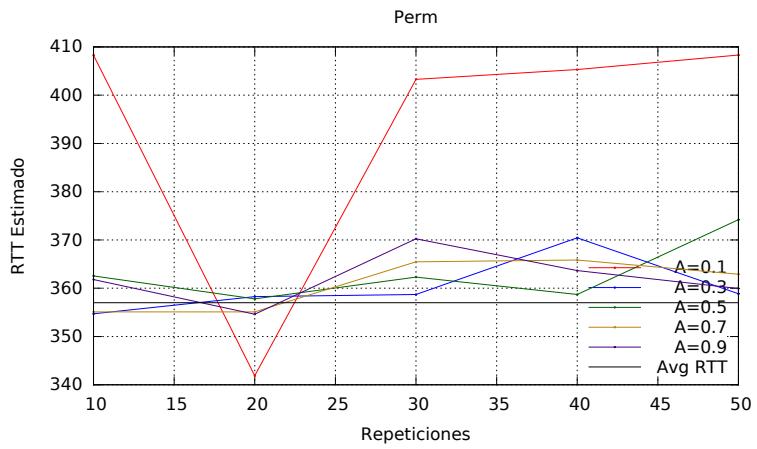
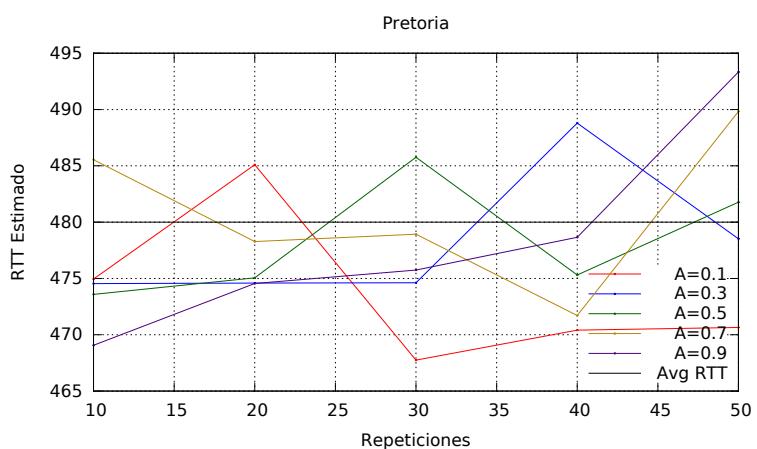
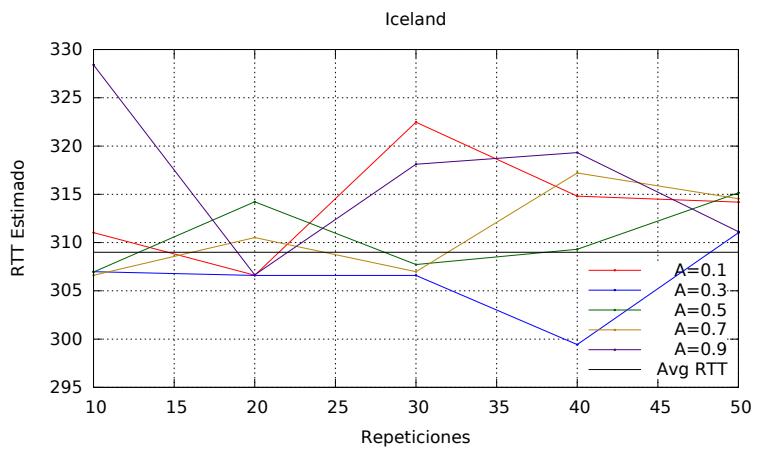
4.2. Estimación de RTT y análisis de parámetros

Antes de comenzar con el detalle de cada universidad debemos justificar la elección de un valor para α y n (cantidad de repeticiones) a usar en el cálculo del *throughput*. Es por esto que incluimos primero esta sección a pesar de corresponder al último experimento realizado. El valor del *throughput* estimado se incluye en el análisis de cada universidad en particular.

En cada gráfico se incluye también una recta con el valor de RTT promedio obtenido en las experimentaciones usando *traceroute*.



Lo primero que notamos es que los resultados parecen ser, dentro de lo razonable, bastante estables: con excepción de Perm, ninguna combinación de parámetros arrojó RTTs distantes entre sí más allá de



45 ms, y todos, con mayor o menor varianza, rondan el resultado obtenido en el experimento de *traceroute*.

También notamos que, como era de esperar, una cantidad mayor de repeticiones parecería dar resultados más cercanos al RTT calculado con el otro método. Pero que, también dentro de lo esperable, valores pequeños de α (es decir, darle mucho peso a las últimas corridas) anulan esta situación.

Decidimos quedarnos pues con $n = 50$. Para ese valor, $\alpha = 0,3$ y $\alpha = 0,5$ parecerían dar los resultados más cercanos al RTT calculado mediante *traceroute*, con lo cual decidimos quedarnos con $\alpha = 0,5$ y usar esos dos parámetros para la estimación del *throughput*.

En todas nuestras pruebas en valor de *MSS* obtenido fue de 1452 bytes, pero lo calculamos programáticamente antes de cada test para asegurarnos de usar el correcto en cada sistema utilizado (ver el método *get_mss* del archivo *main.py*).

Aunque no se incluye en los gráficos la pérdida de paquetes fue realmente baja, llegando a ser nula en la mayoría de las corridas, con lo cual (junto con el *MSS* constante) da como resultado un *throughput* que en los hechos parece depender solamente del *RTT* estimado.

La unidad usada para presentar el *throughput* estimado es B/ms (bytes por milisegundo) que coincide exactamente con kB/s si se usa $kB = 1000B$.

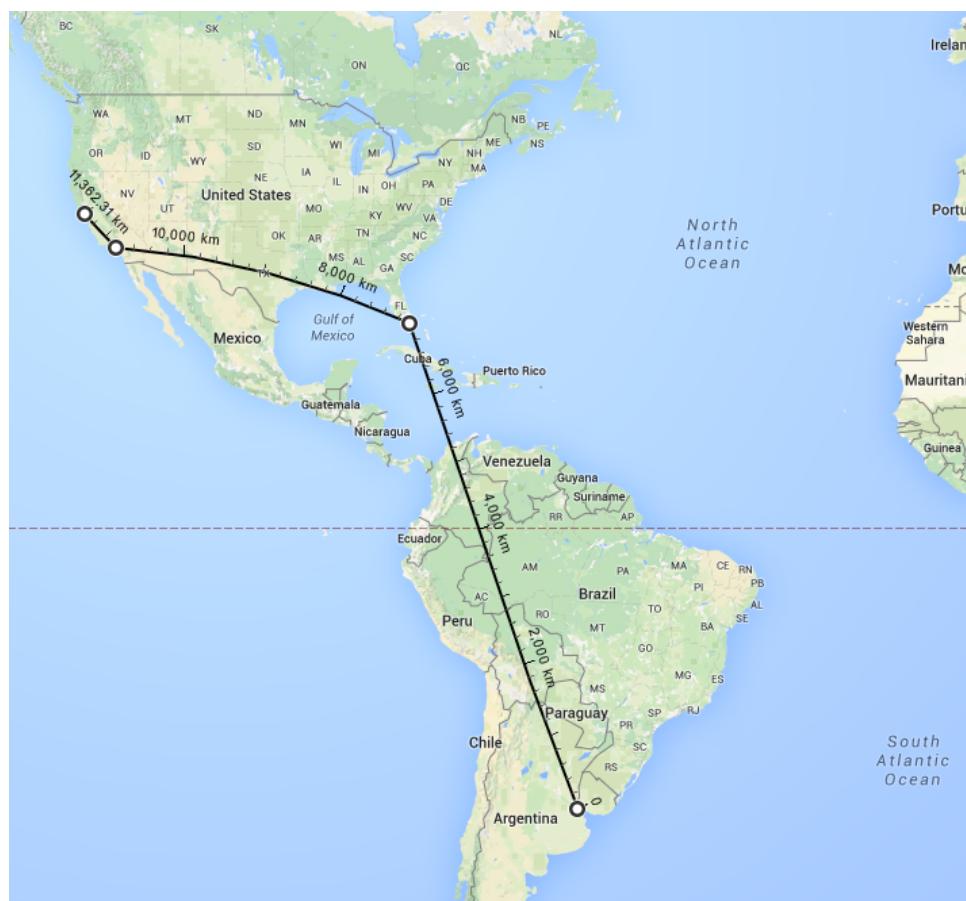
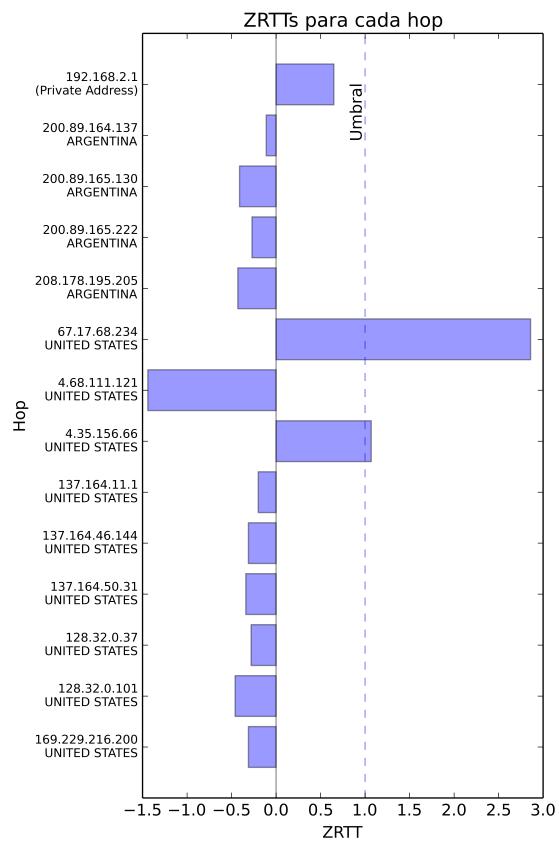
4.3. Berkeley, EEUU

IP	ZRTT	AVG_RTT	PAIS	CIUDAD
192.168.2.1	0.65	35.35	(Private Address)	(Private Address)
200.89.164.137	-0.11	46.40	ARGENTINA	Buenos Aires
200.89.165.130	-0.41	43.80	ARGENTINA	Buenos Aires
200.89.165.222	-0.27	47.29	ARGENTINA	Buenos Aires
208.178.195.205	-0.43	43.57	ARGENTINA	Buenos Aires
67.17.68.234	2.86	191.77	UNITED STATES	Miami
4.68.111.121	-1.44	141.79	UNITED STATES	Miami
4.35.156.66	1.07	207.65	UNITED STATES	Los Angeles
137.164.11.1	-0.2	214.70	UNITED STATES	Los Angeles
137.164.46.144	-0.31	216.72	UNITED STATES	Los Angeles
137.164.50.31	-0.34	217.22	UNITED STATES	Los Angeles
128.32.0.37	-0.28	220.53	UNITED STATES	Berkeley, CA
128.32.0.101	-0.46	215.66	UNITED STATES	Berkeley, CA
169.229.216.200	-0.31	217.43	UNITED STATES	Berkeley, CA

Podemos observar el fenómeno ya mencionado sobre RTTs menores en saltos posteriores. En el gráfico de barras se puede observar que por encima del umbral quedan registrados dos saltos que corresponden a las distancias mas grandes según la geolocalización. Sin embargo, el salto correspondiente al cable trasatlántico entre Buenos Aires y Miami (lugar del llamado NOC de las Américas) queda muy por encima del resto, evidenciando su utilización.

El valor del throughput estimado para esta ruta fue de 6.66 B/ms, el mayor de todas las rutas. Como el throughput está relacionado íntimamente con los valores de RTT, vemos que su alto valor es una consecuencia de la cercanía al objetivo.

```
-- berkeley.edu ping statistics --
100 packets transmitted, 100 received, 0% packet loss, time 99134ms
rtt min/avg/max/mdev = 188.277/190.789/213.277/3.612 ms
```



4.4. Cochin, India

IP	ZRTT	AVG_RTT	PAIS	CIUDAD
192.168.2.1	-0.01	35.71	(Private Address)	(Private Address)
200.89.164.181	-0.33	45.44	ARGENTINA	Buenos Aires
200.89.165.150	-0.43	43.95	ARGENTINA	Buenos Aires
195.22.220.152	-0.42	44.05	ARGENTINA	Buenos Aires
195.22.216.142	1.12	213.52	UNITED STATES	New Orleans
195.22.216.142	-0.43	212.45	UNITED STATES	New Orleans
195.22.195.102	1.59	434.06	ITALY	Milano
218.248.235.161	-1.76	286.69	INDIA	Bangalore
210.212.233.50	1.10	454.65	INDIA	Cochin
210.212.233.50	-0.41	455.32	INDIA	Cochin

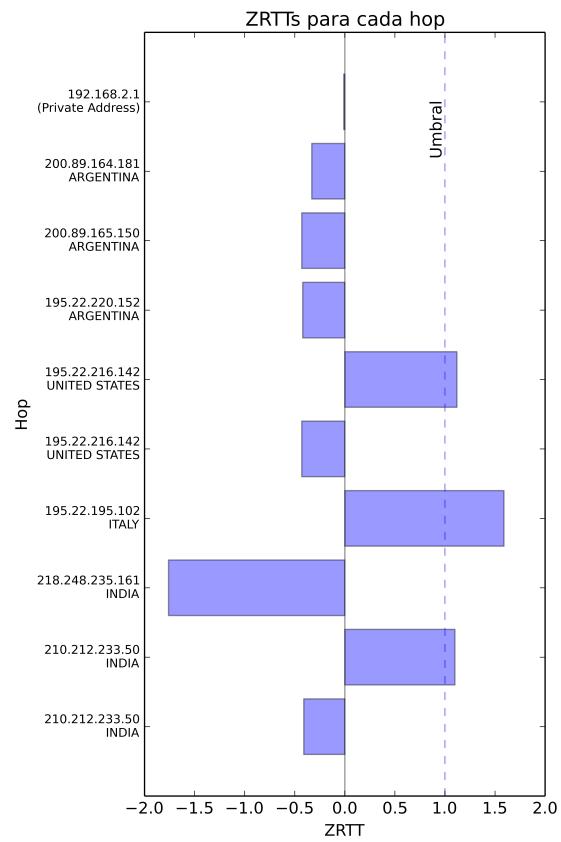
En una primera instancia, notamos que el IP 195.22.220.152 fue marcado como Italiano pero su RTT era demasiado bajo. Verificando contra otras bases de localizacion de IP, comprobamos que efectivamente el nodo se encuentra en Argentina, si bien su rango pertenece a Italia, ya que el ISP dueño es una corporacion Italiana.

Nuevamente vemos una correspondencia entre el salto más largo, trasatlántico esta vez, y la diferencia entre RTTs, reflejada a su vez por el valor de ZRTT. Sin embargo también vemos una diferencia similar entre dos puntos supuestamente cercanos, y una diferencia negativa entre dos puntos supuestamente alejados. Sospechamos que esto último puede ser algún defecto en las bases de datos de geolocalización.

En este caso hay 3 saltos que quedan por encima del umbral, como se puede observar en el gráfico de barras. El primero de ellos corresponde al salto entre Buenos Aires y New Orleans, el segundo al salto entre New Orleans e Italia y el tercero corresponde al salto entre Italia e India. Los primeros dos seguramente involucren el uso de un cable trasatlántico.

El valor del throughput estimado para esta ruta fue de 3.25 B/ms. Comparándolo con el anterior, podemos ver que el valor absoluto mucho mayor de RTT afecta directamente al throughput obtenido.

```
--- cusat.ac.in ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99110ms
rtt min/avg/max/mdev = 420.858/422.976/433.482/1.565 ms
```



4.5. Reykjavik, Iceland

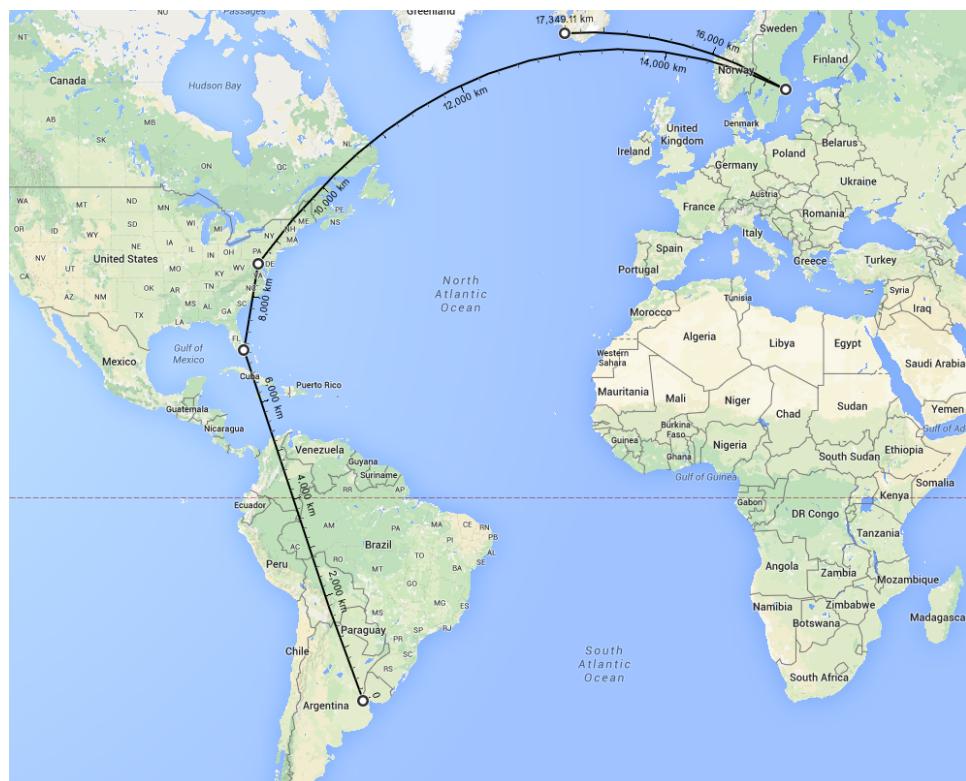
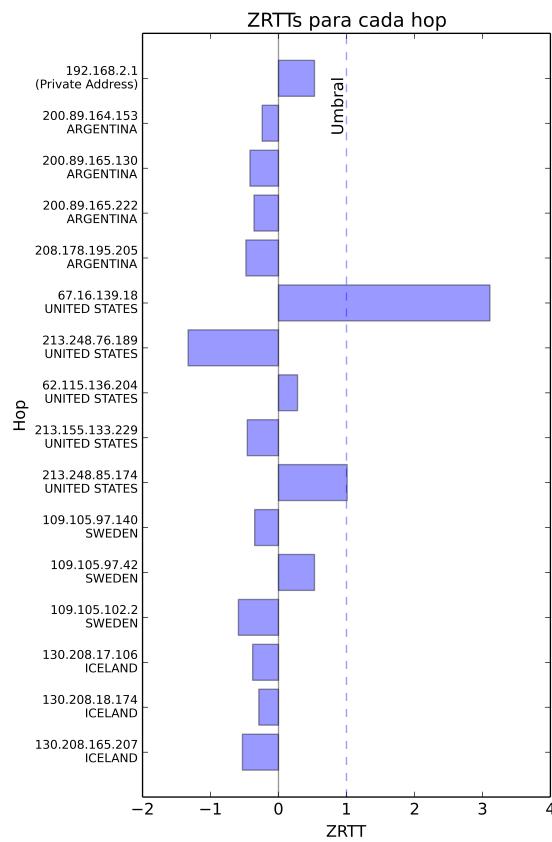
IP	ZRTT	AVG_RTT	PAIS	CIUDAD
192.168.2.1	0.53	37.3	(Private Address)	(Private Address)
200.89.164.153	-0.24	45.36	ARGENTINA	Buenos Aires
200.89.165.130	-0.42	44.90	ARGENTINA	Buenos Aires
200.89.165.222	-0.36	47.18	ARGENTINA	Buenos Aires
208.178.195.205	-0.48	43.77	ARGENTINA	Buenos Aires
67.16.139.18	3.11	211.71	UNITED STATES	Miami
213.248.76.189	-1.33	168.31	UNITED STATES	Miami
62.115.136.204	0.28	201.46	UNITED STATES	Ashburn
213.155.133.229	-0.46	199.20	UNITED STATES	Ashburn
213.248.85.174	1.01	267.33	SWEDEN	Stockholm
109.105.97.140	-0.35	270.48	SWEDEN	Stockholm
109.105.97.42	0.53	315.58	SWEDEN	Stockholm
109.105.102.2	-0.59	307.25	SWEDEN	Stockholm
130.208.17.106	-0.38	308.99	ICELAND	Reykjavik
130.208.18.174	-0.29	314.75	ICELAND	Reykjavik
130.208.165.207	-0.53	309.26	ICELAND	Reykjavik

Una vez más el valor del ZRTT es consistente con el salto grande entre Argentina y EEUU; y con el salto entre Estados Unidos y Suecia. Los IPs 213.155.133.229 y 213.248.85.174 dieron problemas para ser localizados: ambos pertenecen aparentemente a la Unión Europea, pero hay algunos geolocalizadores que les dan una ubicación estadounidense mientras que otros le dan una ubicación sueca. En base a los ZRTT y observando que ciertos IPs pueden tener un rango que no corresponda con su ubicación geográfica, determinamos que el primero está ubicado físicamente en Estados Unidos, mientras que el segundo, en Suecia (o en algún lugar cercano de Europa).

Observando el gráfico de barras correspondiente podemos observar que ambos saltos; tanto el de Buenos Aires a Miami, como el de Ashburn a Stockholm; quedan por encima del umbral elegido. En comparación al enlace hacia la India se puede observar que se toman caminos aparentemente distintos para dar el salto desde Estados Unidos a Europa, ya que este salto posee una diferencia de RTT mucho menor que el visto en la ruta anterior.

El throughput estimado para esta ruta fue de 4.61 B/ms, lo que lo deja ubicado en el medio de los resultados que venimos obteniendo (consistente con nuestro análisis de RTT).

```
--- drupalclvs.rhi.hi.is ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99046ms
rtt min/avg/max/mdev = 274.486/301.449/1920.864/175.128 ms, pipe 2
```



4.6. Perm, Rusia

IP	ZRTT	AVG_RTT	PAIS	CIUDAD
192.168.2.1	0.36	35.15	(Private Address)	(Private Address)
200.89.166.161	-0.30	45.49	ARGENTINA	Buenos Aires
200.89.165.130	-0.51	44.81	ARGENTINA	Buenos Aires
200.89.165.222	-0.46	46.45	ARGENTINA	Buenos Aires
208.178.244.213	-0.52	45.05	ARGENTINA	Buenos Aires
67.17.75.66	2.54	205.42	UNITED STATES	Miami
4.68.111.121	-1.06	175.47	UNITED STATES	Miami
4.69.158.245	1.89	301.85	SWEDEN	Stockholm
4.69.158.245	-0.42	305.45	SWEDEN	Stockholm
213.242.110.198	-0.26	317.85	SWEDEN	Stockholm
194.85.40.229	-0.29	328.49	RUSSIAN FEDERATION	Saint Petersburg
194.226.194.22	0.01	355.20	RUSSIAN FEDERATION	Saint Petersburg
212.192.80.57	-0.57	351.03	RUSSIAN FEDERATION	Perm
212.192.64.44	-0.37	357.50	RUSSIAN FEDERATION	Perm

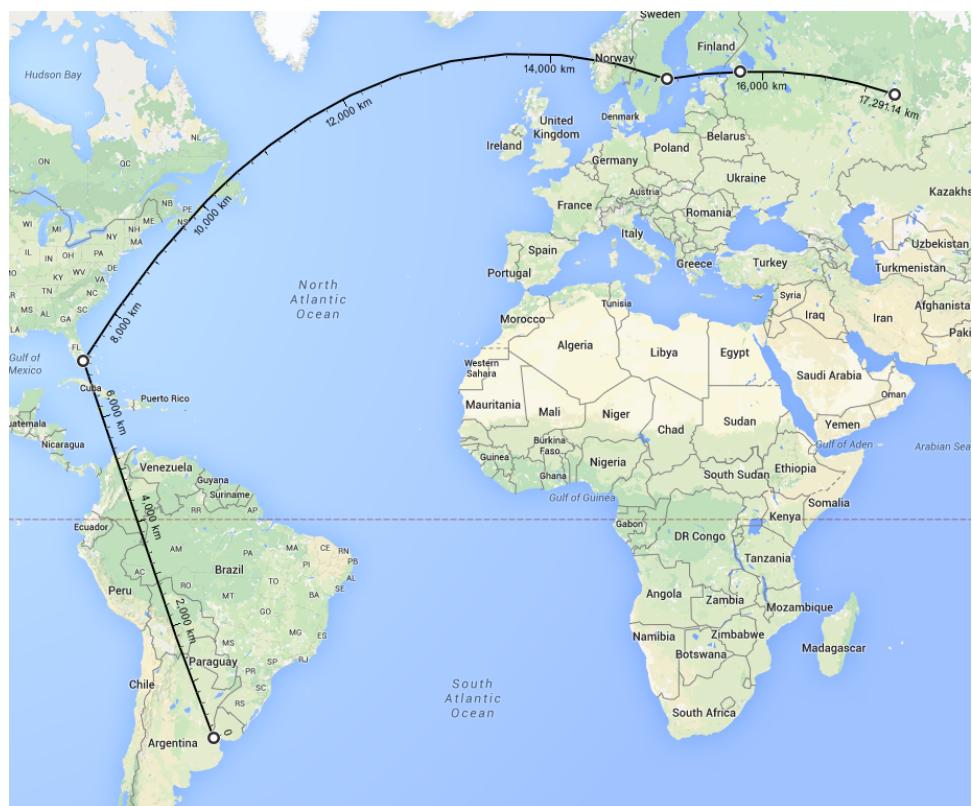
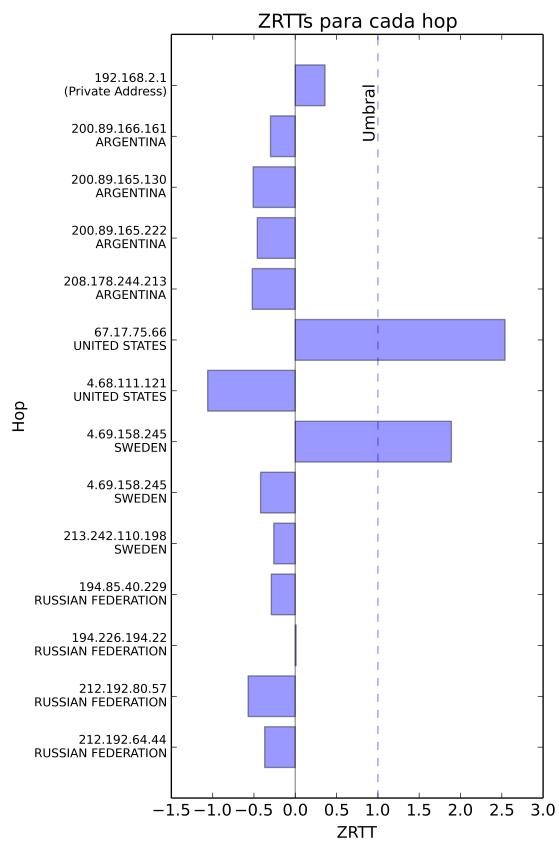
Una vez más se debe transitar por medio de Miami para llegar a Europa, y como se puede observar el ZRTT es consistente con este salto.

Inicialmente esta ruta fue complicada de analizar, ya que obteníamos un salto transatlántico sorprendentemente corto. Luego de corroborar mejor nuestras fuentes de geolocalización IP lo mismo cobró más coherencia observando que el salto transatlántico entre Estados Unidos y Suecia era lo que tenía el ZRTT alto. Una vez más en el gráfico de barras podemos observar dos saltos por encima del umbral, correspondientes al salto entre Buenos Aires y Miami y al discutido recientemente.

Esta ruta fue la única ruta para la que obtuvimos pérdida de paquetes, obteniendo una *EstimatedPacketLossProbability* = 0,02. En los demás casos este valor no se incluye por ser siempre = 0.

El throughput estimado para esta ruta fue de 3.82 B/ms, un valor comparativamente bajo aunque no tanto como el enlace con India. La pequeña probabilidad de paquetes no pareciera alterar en gran medida nuestro análisis sobre la importancia del RTT en su cálculo.

```
--- psu.ru ping statistics ---
100 packets transmitted, 98 received, 2% packet loss, time 99050ms
rtt min/avg/max/mdev = 324.972/333.262/347.473/6.723 ms
```



4.7. Pretoria, South Africa

IP	ZRTT	AVG_RTT	PAIS	CIUDAD
192.168.2.1	0.42	35.29	(Private Address)	(Private Address)
200.89.164.177	-0.20	45.07	ARGENTINA	Buenos Aires
200.89.165.130	-0.38	44.64	ARGENTINA	Buenos Aires
200.89.165.222	-0.36	45.49	ARGENTINA	Buenos Aires
208.178.195.205	-0.43	42.76	ARGENTINA	Buenos Aires
67.17.106.162	2.64	211.71	UNITED STATES	Miami
154.54.13.61	-1.14	168.90	UNITED STATES	Atlanta
154.54.24.233	-0.37	169.24	UNITED STATES	Atlanta
154.54.24.197	-0.13	183.11	UNITED STATES	Atlanta
154.54.31.110	-0.20	193.00	UNITED STATES	Atlanta
154.54.7.26	-0.28	198.41	UNITED STATES	Chicago
154.54.31.118	-0.39	197.44	UNITED STATES	New York
154.54.30.186	1.13	282.38	UNITED KINGDOM	London
130.117.50.201	-0.44	278.99	ITALY	Milan
154.54.38.190	-0.22	287.66	UNITED KINGDOM	London
149.14.80.210	-0.01	308.23	UNITED KINGDOM	London
196.32.209.50	-0.66	292.33	SOUTH AFRICA	Cape Town
196.32.209.117	3.08	485.75	SOUTH AFRICA	Cape Town
155.232.6.86	-0.63	471.57	SOUTH AFRICA	Wynberg
155.232.6.29	-0.07	488.65	SOUTH AFRICA	Wynberg
155.232.6.138	-0.66	472.79	SOUTH AFRICA	Wynberg
137.215.99.2	-0.17	484.43	SOUTH AFRICA	Pretoria
137.215.10.70	-0.45	480.05	SOUTH AFRICA	Pretoria

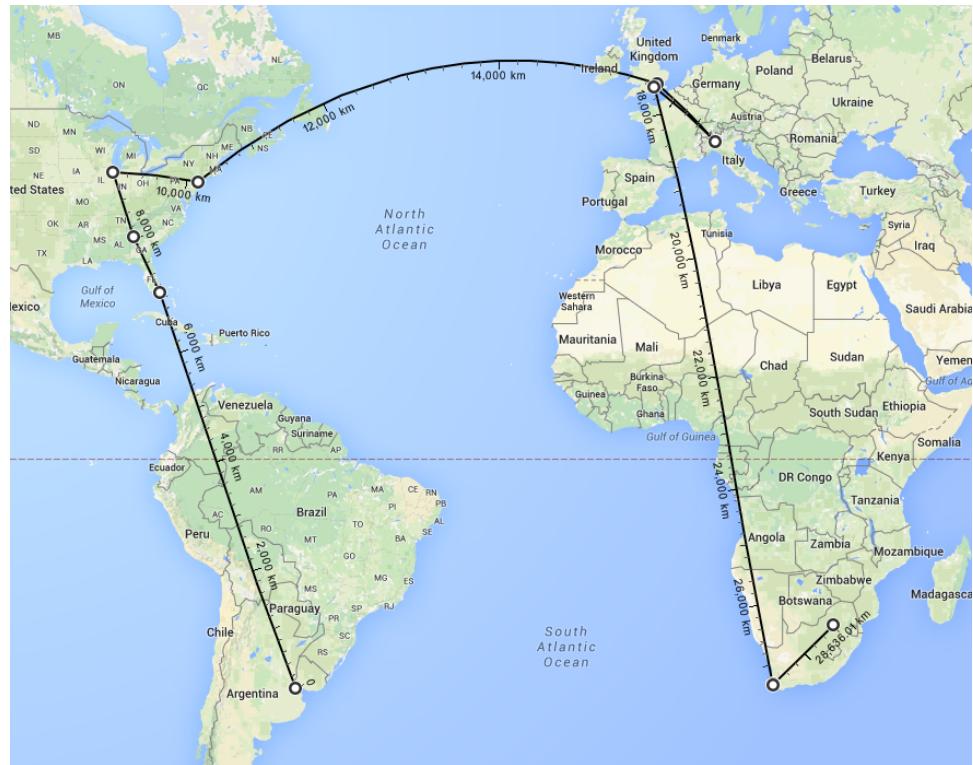
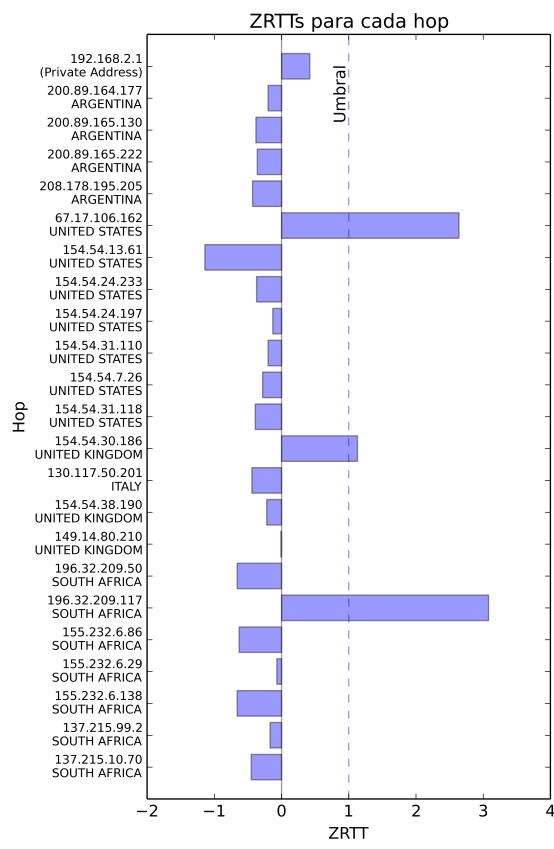
Por lejos nuestra ruta más larga, lo primero que notamos es que a pesar de la relativa cercanía geográfica, para conectar Sudamérica con África hace falta pasar por continentes del norte (y una vez más por Miami, casi una constante de todas las rutas elegidas). Esto puede deberse a la ausencia de un cable trasatlántico que conecte directamente Sudamérica con África. Esta vez el salto trasatlántico sí tiene un ZRTT grande y el otro salto significativo es dentro de Cape Town, en Sudáfrica, lo que nos resultó llamativo.

Creemos que este último puede tener que ver con las demoras de encolamiento (propias para paquetes ICMP) o situaciones similares en loops locales de Sudáfrica.

Además, hemos re-analizado la diferencia entre los hosts 196.32.209.50 y 196.32.209.117 por separado, ya que ambos responden al ping, con herramientas adicionales (comando *ping* de Linux) como para verificar mejor esta situación puntualmente. Estos checkeos nos dieron resultados similares a los calculados por nuestra herramienta (aproximadamente 175ms de diferencia).

El throughput estimado para esta ruta fue de 3.03 B/ms, el más bajo de las rutas usadas. Entendemos que tiene que ver con la gran distancia física que recorre la conexión trazada y su consiguiente RTT mayor a las demás rutas.

```
--> up.ac.za ping statistics --
100 packets transmitted, 100 received, 0% packet loss, time 99104ms
rtt min/avg/max/mdev = 446.330/452.868/510.654/10.853 ms
```



5. Conclusiones

Pudimos comprobar empíricamente la relación directamente proporcional entre el RTT y la distancia geográfica de las rutas, y su relación inversa con el valor estimado por el throughput obtenido por la ecuación de Mathis.

El umbral propuesto pareciera identificar correctamente los enlaces submarinos que conectan al mundo entre sí, pero también arroja siempre como “falso positivo” el enlace Buenos Aires - Miami. De todos modos, dada su gran distancia y la geografía de América, es probable que dicho enlace sea también submarino (aunque seguramente no sea trasatlántico) por lo que no descartamos de lleno el método propuesto para identificarlos.

En el intento de medir hasta qué host llega el paquete ICMP con determinado TTL o si llega a destino, nos encontramos con que algunos de los hosts no respondían nada. Esto resulta de la configuración de los mismos que por medidas de seguridad no permiten, o bien responder paquetes o bien filtran paquetes ICMP. Estos hosts no fueron incluidos en el análisis de las rutas y los RTT finales de cada hop incluyen el tiempo que toma transitar por Hosts que no registran estos paquetes. Como conclusión a esto, hoy en día el uso de los paquetes ICMP para realizar traceroutes no parece fiable, dado que no todos los routers están configurados para tratar con ellos. Por esta razón creemos que utilizar paquetes TCP sería de mayor precisión para obtener todos los hops de una ruta, aunque habría que constatarlo experimentalmente.

Nos encontramos con resultados interesantes cuando analizamos los datos, por ejemplo cuando intentamos realizar la geolocalización de los distintos IPs. Existen IPs que tienen un rango correspondiente a un país específico pero físicamente están ubicados en otras regiones del mundo, lo que dificultó la ubicación de los mismos y la creación de los distintos mapas. Por otro lado, mediante la detección de cambios de ruta, pudimos observar que países como India tienen una alta redundancia en sus rutas (o un problema en su red), ya que se detectaron muchos cambios de rutas al realizar las repeticiones. Además, notamos que había muchas universidades que no responden a un echo-request (por ejemplo la Universidad de Tokyo, Japón) por lo que se dificultó su selección.

En un trabajo futuro, sería buena idea analizar las distintas rutas que puede tomar un paquete para llegar a destino, exponiendo los lugares por los que transita y los tiempos de cada uno de los caminos. También queda pendiente realizar este mismo análisis y otros utilizando paquetes TCP con algún puerto que sepamos que se encuentra abierto hasta el host de destino para mayor precisión.