Teoría de las Comunicaciones
TP2

8 de junio de 2016

Integrante	LU	Correo electrónico
Martín Baigorria	575/14	martinbaigorria@gmail.com
Federico Beuter	827/13	federicobeuter@gmail.com
Mauro Cherubini	835/13	cheru.mf@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción

Cuando un usuario accede a un sitio web o intenta transmitir información de un punto a otro por medio de la web, el mismo en general se abstrae de todos los mecanismos necesarios requeridos para esta transmisión. Dada la gran cantidad y heterogeneidad del hardware subyacente, en lo comienzos de la web se empezaron a diseñar distintos tipos de protocolos de comunicación para lograr unificar todos estos dispositivos en una red capaz de transmitir información desde cualquier punto a otro de forma relativamente confiable.

Hoy en día, el protocolo dominante para la transmisión de datos por la red es el IP (Internet Protocol). Creado por Vint Cerf y Bob Kahn en 1974, el protocolo IP esta diseñado para ser utilizado en redes de conmutación de paquetes y no esta orientado a conexión. Esto significa que al enviar un archivo, el mismo es fragmentado en paquetes que no necesariamente siguen la misma ruta en la red. Además, al no ser un protocolo orientado a conexión, no hay garantías de que los paquetes lleguen a destino dado que no hay ningún tipo de protocolo de handshake entre origen y destino.

En relación al modelo OSI, el protocolo IP pertenece a la capa de internet. Existen muchos otros protocolos que han sido construidos sobre el protocolo IP con el objetivo de proveer otro tipo de garantías y funcionalidades, entre los mas conocidos se encuentran el protocolo TCP (Transmission Control Protocol) y el UDP (User Datagram Protocol). Estos protocolos pertenecen a la capa de interconexión.

En el presente trabajo utilizaremos el protocolo ICMP (Internet Control Message Protocol), el cual esta especificado en el RFC 792 [3], con el objetivo de analizar las diferentes trazas y el RTT (round trip time) al momento de conectarse a un sitio web. Una traza se define como la sucesión de dispositivos de red que son recorridos, ya sean puentes, routers o gateways, al momento de transmitir información en la red. El protocolo ICMP esta implementado sobre IP, aunque se considera que el mismo no pertenece a la capa de interconexión a diferencia de TCP y UDP. Esto se debe a que su principal propósito no es intercambiar datos entre sistemas, si no que en general se utilizan para enviar mensajes de error entre dispositivos de red, con la excepción de su uso en herramientas como ping y traceroute.

1.1. Traceroute

La herramienta traceroute, desarrollada inicialmente por Van Jacobson en 1988, es una herramienta sumamente útil de diagnostico de red para buscar una aproximación de la traza de conexión y encontrar el RTT a cada hop (o salto) en la traza. Como ya hemos mencionado, esta herramienta utiliza el protocolo ICMP. A continuación discutiremos la estructura de los paquetes IP/ICMP para luego explicar como se hace efectivamente para identificar los diferentes hops. Luego discutiremos que potenciales problemas puede tener esta herramienta.

1.2. Header IP

Como ya hemos mencionado, el protocolo ICMP esta implementado sobre IP. A continuación mostramos la estructura del header de un paquete IPv4. Todo lo que mencionaremos sigue siendo valido para IPv6.

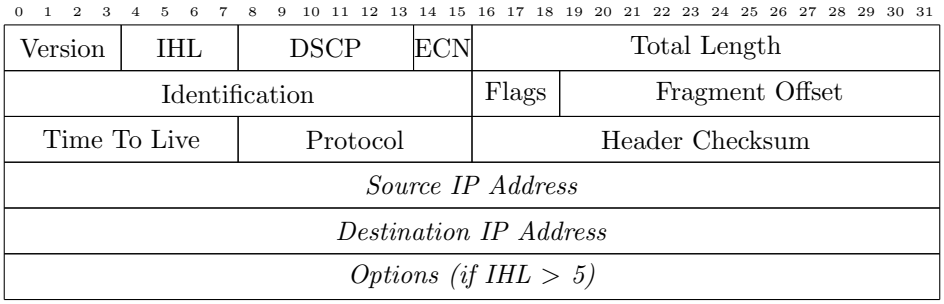


Figura 1: Header IPv4

Como podemos observador, el header de un paquete IPv4 tiene 24 bytes. En este momento,

el campo que mas nos interesa es Time To Live (TTL). Como lo dice su nombre, este campo fue pensado para imponer un limite de tiempo a la vida del paquete en la red. Si no llegaba el paquete antes de ese tiempo, el mismo era descartado por el correspondiente hop. Sin embargo, en la practica esto se implemento como un limite a la cantidad de hops que un paquete podía recorrer. Esto se ve en los headers de los paquetes IPv6, donde el campo fue renombrado como Hop Limit.

1.3. Header ICMP

En los paquetes IP/ICMP, al header de IP se le suma el header de ICMP. El mismo tiene la siguiente estructura:

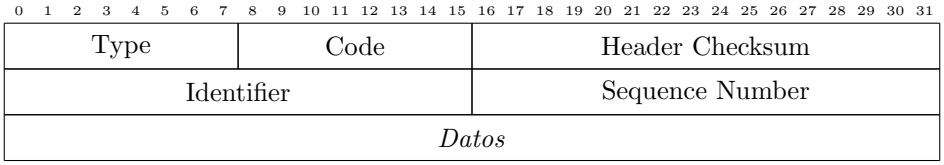


Figura 2: Header ICMP

El protocolo ICMP es parte del Internet Protocol Suite, especificado en RFC 792. Este header de 12 bytes se ubica luego del header de IP, teniendo un tamaño de header total de 36 bytes. La especificación explica en detalle como se utiliza el protocolo normalmente. Lo bueno de este protocolo es que si en algún momento header de IP llega a un $ttl = 0$, el hop correspondiente devolverá un mensaje de error al cliente de origen. En unos momentos veremos porque esto es sumamente útil.

En nuestro caso, los siguientes casos de type seran los mas relevantes:

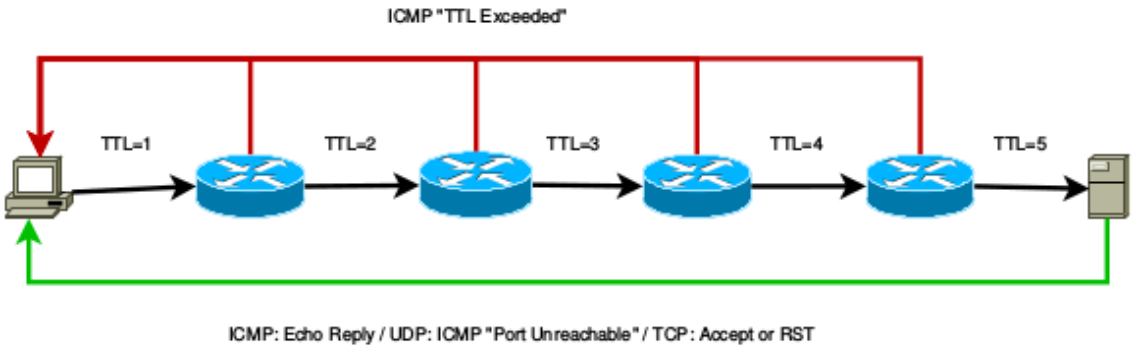
- Type 0 – Echo Reply
- Type 3 - Destination Unreachable
- Type 8 – Echo Request
- Type 11 – Time Exceeded

En principio, implementaremos nuestra propia herramienta traceroute utilizando paquetes ICMP. Para ello, enviaremos un Echo Request al URL (Uniform Resource Locator) al que queremos acceder y encontrar la traza utilizando el campo TTL del header IP.

Si el paquete llega al destino, el servidor nos enviara un paquete ICMP con el type Echo Reply. Para poder encontrar los diferentes hops en la traza, utilizaremos el parámetro Time To Live del header IP, inicializándolo en 1 y luego aumentándolo hasta que nos respondan con un Echo Reply. Es decir, si estamos interesados en identificar el hop i de la traza, tendremos que settear el parametro $TTL = i$ y luego enviar el request.

Cuando TTL llega a 0, si el hop implementa ICMP, el mismo nos devolverá un reply de tipo Time Exceeded, incluyendo en la sección de datos del paquete ICMP el header IP y los primeros 8 bytes de del datagrama de datos original que enviamos. Si el hop se encuentra bajo algún tipo de congestión, es posible que el mismo descarte nuestro paquete para priorizar los de protocolos como TCP o UDP. Esto por supuesto depende de la implementación del dispositivo de red correspondiente.

Notar que no necesariamente el servidor estara disponible o aceptara paquetes ICMP, por lo que tendremos que poner un limite a la cantidad de hops que buscaremos. Caso contrario iteraríamos hasta llegar al limite dado por los 8 bytes del campo TTL, lo cual no tiene sentido practico. Este procedimiento se puede ver un poco mejor en la siguiente imagen:



Para implementar esta herramienta utilizaremos Python, con la libreria Scapy. Esta librería nos permite formar paquetes ICMP y luego hacer los respectivos requests.

1.4. Resolución DNS

Al hacer un request, en general nos abstraemos de la direccion IP y lo hacemos a un URL. Este URL se debe resolver a una dirección IP mediante un request a un DNS (Domain Name System) con el formato especificado en el RFC 1035 [1]. Por lo tanto, correr dos veces la herramienta no nos garantizara que hagamos el request a un mismo IP, dado que un sitio puede tener varios IPs asignados. Esto pasa normalmente con google.com. A su vez, en el ejemplo ilustrativo consideramos una topología de red sumamente simple. Dado que las topologías tienden a ser sumamente complejas, esto lleva a que al hacer el traceroute se puedan presentar una serie de problemas que deben ser tenidos en cuenta.

1.5. Anomalidades en traceroutes

A continuación veremos las potenciales problemáticas de hacer un traceroute utilizando paquetes ICMP. En general las mismas surgen debido a la complejidad innata de las topologías de red. Las mismas en general se pueden agrupar en los siguientes tipos:

1. Missing hops
2. Missing destination
3. False RRTs
4. Missing links
5. Loops and Circles
6. Diamonds

Estos tipos están explicados en detalle en el paper de Jobst [4].

[1] RFC 1035: `https://www.ietf.org/rfc/rfc1035.txt`

2. Discusión

2.1. Traceroute a Universidades

A continuación mostramos diferentes traceroutes a universidades en diferentes continentes. Consideramos que esto es relevante dado que nos permitirá observar enlaces intercontinentales y otras particularidades. El RTT promedio se calculo a partir de 5 requests, tomando como tiempo inicial y final el tiempo dado por los paquetes IP. No utilizamos el tiempo de maquina dado que el mismo sesga las mediciones al agregar el overhead del lenguaje que utilizamos. Notar a su vez que el RTT no sera necesariamente creciente, dado que es una variable sujeta al ruido de la red. Esto significa que es posible que al calcular el tiempo de un enlace, el mismo nos de negativo. El host name lo conseguimos a partir de la función socket.gethostbyaddr de Python, que lo que hace es simplemente un DNS lookup [2]. La ubicación la conseguimos a partir de una base de datos publica de GeoIP.

2.1.1. Detección de enlaces intercontinentales

Método de Simbala

Para detectar enlaces intercontinentales, utilizaremos el método de Simbala [2] para detectar outliers sobre los tiempos entre hops, es decir, tiempo de enlace. Cuando no observamos el RTT, ignoraremos el respectivo enlace. El procedimiento se encuentra claramente explicado en el apunte, básicamente es un test con una región de rechazo. Diremos que si el test identifica al enlace como un outlier, el mismo es un enlace intercontinental.

GeoIP

Por otro lado, utilizaremos mediante Python la base de datos de GeoIP. Si GeoIP muestra que el origen y el destino del enlace pertenecen a diferentes continentes, el mismo sera un enlace intercontinental. Notar que la base de datos GeoIP puede fallar, por lo que también tendremos falsos positivos y negativos.

2.1.2. dc.ubar.ar

Cuadro 1: traceroute: dc.uba.ar

Hop	Avg. RTT	IP Address	Host name	Location
1	9.3842 ms	181.169.12.1	1-12-169-181.fibertel.com.ar	AR, SA
2	* * * * *			
3	* * * * *			
4	* * * * *			
5	14.025 ms	200.89.164.53	53-164-89-200.fibertel.com.ar	AR, SA
6	14.7514 ms	200.89.165.2	2-165-89-200.fibertel.com.ar	AR, SA
7	22.5916 ms	200.89.165.86	86-165-89-200.fibertel.com.ar	AR, SA
8	16.5408 ms	200.49.69.161	VPN-corp.metrored.net.ar	AR, SA
9	* * * * *			
10	* * * * *			
11	* * * * *			
12	12.7052 ms	157.92.47.53	157.92.47.53	AR, SA
13	13.067 ms	192.168.121.2	192.168.121.2	
14	* * * * *			
...	* * * * *			

Como es de esperar, al hacer un request a dc.uba.ar desde Argentina no hay saltos intercontinentales. Sin embargo, notemos que este traceroute cae en el problema de *missing destination*. Esto no es porque el servidor no exista, si no porque probablemente esta configurado para no devolver ICMP requests.

[2] El manpage de la libc explica como se hace esto `http://www.freebsd.org/cgi/man.cgi?query=gethostbyaddr&sektion=3&manpath=FreeBSD+6.0-RELEASE`

A su vez, en este trabajo y en este caso en particular las trazas en general muestran una serie de *missing hops*. Esto en general se debe a las configuraciones de los dispositivos de red, que no aceptan paquetes ICMP.

Intentamos acceder a metrored.net.ar, ya sea por URL o por IP y no lo logramos. Sin embargo, al hacer un IP Whois encontramos:

Cuadro 2: Whois lookup: metrored.net.ar

owner:	Techtel LMDS Comunicaciones Interactivas S.A.
ownerid:	AR-TLCI-LACNIC
responsible:	Administrador de Direcciones IP - CLARO
address:	Garay, 34
address:	C1063AB - Buenos Aires
country:	AR
phone:	+54 11 4000-3000 [3270]
nserver:	DNSMR1.METRORED.NET.AR

Por lo que podemos ver, el hop pertenece a Claro.

2.1.3. mit.edu

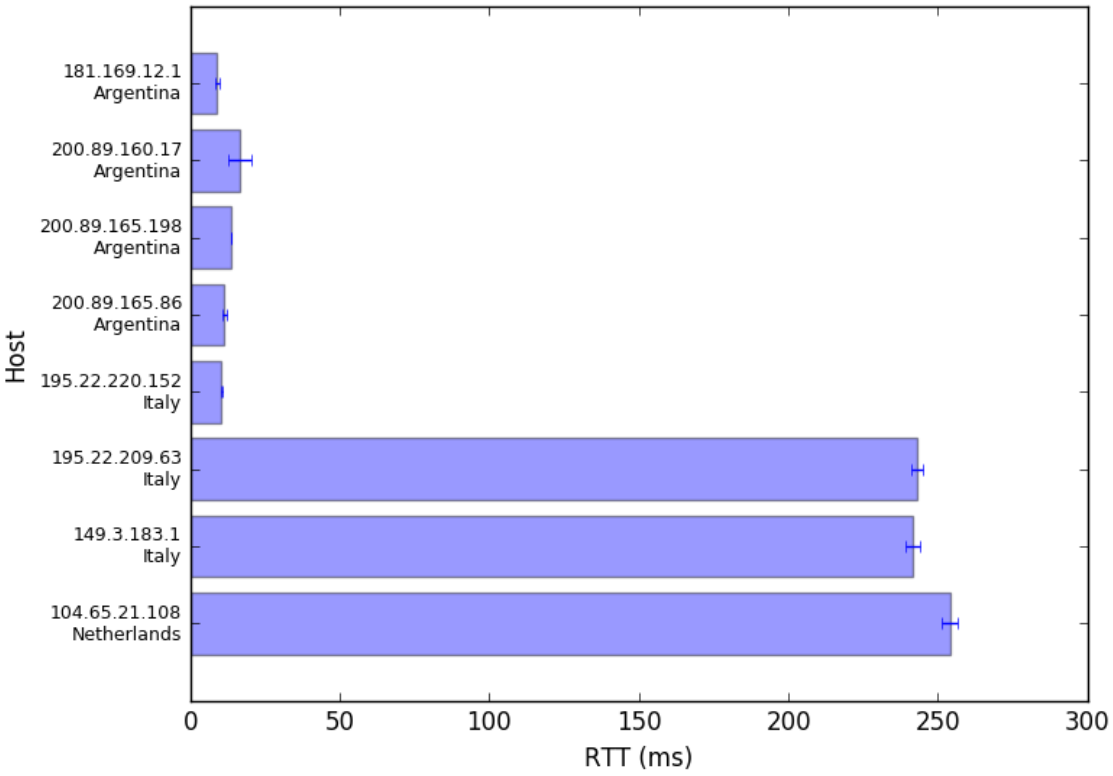
Cuadro 3: traceroute: mit.edu

Hop	Avg. RTT	IP Address	Host name	Location
1	8.987 ms	181.169.12.1	1-12-169-181.fibertel.com.ar	Argentina
2	* * * * *			
3	* * * * *			
4	* * * * *			
5	16.4364 ms	200.89.160.17	17-160-89-200.fibertel.com.ar	Argentina
6	13.6058 ms	200.89.165.198	198-165-89-200.fibertel.com.ar	Argentina
7	11.3204 ms	200.89.165.86	86-165-89-200.fibertel.com.ar	Argentina
8	10.5236 ms	195.22.220.152	xe-1-2-1.baires3.bai.seabone.net	Italy
9	242.916 ms	195.22.209.63	et-10-1-0.londra32.lon.seabone.net	Italy
10	241.5702 ms	149.3.183.1		Italy
11	253.9788 ms	104.65.21.108	a104-65-21-108.deploy.static.akamaitechnologies.com	Netherlands

En este gráfico hay un enlace transatlántico claramente identificable entre el hop 7 y el hop 8. Notar que el host name ya indica que es transatlántico. Buscando a quien pertenece seabone.net, averiguamos que pertenece a la empresa Sparkle que provee servicios de enlaces transatlánticos.

> Sparkle is a leading global telecommunication service provider, offering a complete range of IP, Data, Cloud, Data Center, Mobile and Voice solutions designed to meet the ever changing needs of Fixed and Mobile Operators, ISPs, OTTs, Media & Content Players, Application Service Providers and Multinational Corporations (MNCs)

Figura 3: RTTs de los hosts hasta mit.edu



Curiosamente, al final el request termino en Holanda en nodo de Akamai y no en Estados Unidos. Haciendo un Whois al URL confirmamos que esto es correcto.

Cuadro 4: Detección de enlaces intercontinentales: mit.edu

Id	From	To	Avg. RTT	Simbala	GeoIP
1	181.169.12.1 Argentina, SA fibertel.com.ar	200.89.160.17 Argentina, SA fibertel.com.ar	7.4494 ms	no	no
2	195.22.220.152 Italy, EU bai.seabone.net	195.22.209.63 Italy, EU lon.seabone.net	232.3924 ms	yes	no
3	149.3.183.1 Italy, EU	104.65.21.108 Netherlands, EU static.akamaitechnologies.com	12.4086 ms	no	no

Como podemos observar, el método de Simbala detecto un enlace intercontinental. La base de datos de GeoIP ubico al IP source en Italia. Sin embargo el hostname da evidencia a favor de que el mismo en realidad se encuentra en Buenos Aires. Los sitios de geolocalizacion online confirman esto. Es decir, el enlace 2 efectivamente es intercontinental.

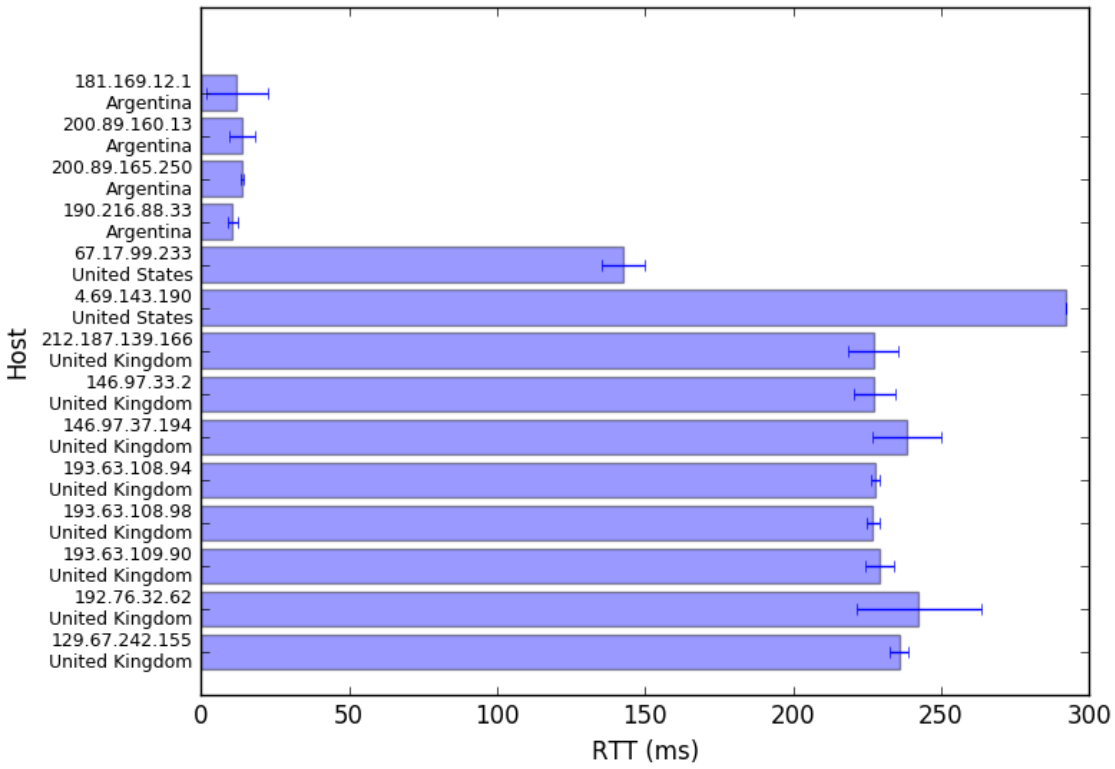
2.1.4. ox.ac.uk

Cuadro 5: traceroute: ox.ac.uk (oxford)

Hop	Avg. RTT	IP Address	Host name	Location
1	12.1034 ms	181.169.12.1	1-12-169-181.fibertel.com.ar	Argentina
2	* * * * *			
3	* * * * *			
4	* * * * *			
5	13.949 ms	200.89.160.13	13-160-89-200.fibertel.com.ar	Argentina
6	13.857 ms	200.89.165.250	250-165-89-200.fibertel.com.ar	Argentina
7	* * *			
7	10.7125 ms	190.216.88.33		Argentina
8	142.5634 ms	67.17.99.233	ae0-300G.ar5.MIA1.gblx.net	United States
9	* * * * *			
10	* * * *			
10	292.234 ms	4.69.143.190	ae-1-3104.ear2.London2.Level3.net	United States
11	227.1146 ms	212.187.139.166	unknown.Level3.net	United Kingdom
12	227.4534 ms	146.97.33.2	ae29.londpg-sbr2.ja.net	United Kingdom
13	238.4376 ms	146.97.37.194	ae19.readdy-rbr1.ja.net	United Kingdom
14	227.6364 ms	193.63.108.94	ae2.oxfoii-rbr1.ja.net	United Kingdom
15	226.9026 ms	193.63.108.98	ae3.oxforq-rbr1.ja.net	United Kingdom
16	229.074 ms	193.63.109.90	oxford-university.ja.net	United Kingdom
17	* * * * *			
18	* * * * *			
19	242.347 ms	192.76.32.62	boucs-lompi1.sdc.ox.ac.uk	United Kingdom
20	235.7944 ms	129.67.242.155	aurochs-web-155.nsms.ox.ac.uk	United Kingdom

Aquí podemos identificar claramente enlaces transatlánticos a partir del host name y la ubicación. Level3 es una empresa conocida proveedora de enlaces. Un dato de color, sus acciones cotizan en Nasdaq.

Figura 4: RTTs de los hosts hasta ox.ac.uk



Cuadro 6: Detección de enlaces intercontinentales: ox.ac.uk

Id	From	To	Avg. RTT	Simbala	GeoIP
1	181.169.12.1 Argentina, SA fibertel.com.ar	200.89.160.13 Argentina, SA fibertel.com.ar	1.8456 ms	no	no
2	200.89.160.13 Argentina, SA fibertel.com.ar	200.89.165.250 Argentina, SA fibertel.com.ar	-	no	no
3	200.89.165.250 Argentina, SA fibertel.com.ar	190.216.88.33 Argentina, SA	-	no	no
4	190.216.88.33 Argentina, SA	67.17.99.233 United States, NA MIA1.gblx.net	131.8509 ms	yes	yes
5	67.17.99.233 United States, NA MIA1.gblx.net	4.69.143.190 United States, NA London2.Level3.net	149.6706 ms	yes	no
6	4.69.143.190 United States, NA London2.Level3.net	212.187.139.166 United Kingdom, EU unknown.Level3.net	-	no	yes
7	212.187.139.166 United Kingdom, EU unknown.Level3.net	146.97.33.2 United Kingdom, EU londpg-sbr2.ja.net	0.3388 ms	no	no
8	146.97.33.2 United Kingdom, EU londpg-sbr2.ja.net	146.97.37.194 United Kingdom, EU readdy-rbr1.ja.net	10.9842 ms	no	no
9	146.97.37.194 United Kingdom, EU readdy-rbr1.ja.net	193.63.108.94 United Kingdom, EU oxfoii-rbr1.ja.net	-	no	no
10	193.63.108.94 United Kingdom, EU oxfoii-rbr1.ja.net	193.63.108.98 United Kingdom, EU oxforq-rbr1.ja.net	-	no	no
11	193.63.108.98 United Kingdom, EU oxforq-rbr1.ja.net	193.63.109.90 United Kingdom, EU oxford-university.ja.net	2.1714 ms	no	no
12	193.63.109.90 United Kingdom, EU oxford-university.ja.net	192.76.32.62 United Kingdom, EU ox.ac.uk	13.273 ms	no	no
13	192.76.32.62 United Kingdom, EU ox.ac.uk	129.67.242.155 United Kingdom, EU ox.ac.uk	-	no	no

El enlace numero 4 claramente es intercontinental. Nuevamente nos encontramos con un error de la base de datos de GeoIP en el enlace numero 5, que también es intercontinental aunque el nodo de destino esta etiquetado incorrectamente. En el enlace numero 6 no tenemos un RTT pero GeoIP lo clasifica correctamente como intercontinental.

2.1.5. u-tokyo.ac.jp

Cuadro 7: traceroute: u-tokyo.ac.jp

Hop	Avg. RTT	IP Address	Host name	Location
1	10.9274 ms	181.169.12.1	1-12-169-181.fibertel.com.ar	Argentina
2	* * * * *			
3	* * * * *			
4	* * * * *			
5	14.2606 ms	200.89.160.21	21-160-89-200.fibertel.com.ar	Argentina
6	12.1018 ms	200.89.165.222	222-165-89-200.fibertel.com.ar	Argentina
7	12.8424 ms	195.22.220.102	xe-1-0-3.baires5.bai.seabone.net	Italy
8	40.0902 ms	195.22.219.17	ae7.sanpaolo8.spa.seabone.net	Italy
9	39.1184 ms	195.22.219.17	ae7.sanpaolo8.spa.seabone.net	Italy
10	41.755 ms	149.3.181.65		Italy
11	163.8182 ms	129.250.2.227	ae-4.r24.nycmny01.us.bb.gin.ntt.net	United States
12	226.2194 ms	129.250.4.13	ae-2.r20.sttlwa01.us.bb.gin.ntt.net	United States
13	237.003 ms	129.250.2.54	ae-0.r21.sttlwa01.us.bb.gin.ntt.net	United States
14	387.993 ms	129.250.3.86	ae-2.r20.osakjp02.jp.bb.gin.ntt.net	United States
15	403.096 ms	129.250.6.188	ae-4.r22.osakjp02.jp.bb.gin.ntt.net	United States
16	382.5294 ms	129.250.2.255	ae-1.r01.osakjp02.jp.bb.gin.ntt.net	United States
17	393.0274 ms	61.200.80.218	xe-0-4-0-7.r01.osakjp02.jp.ce.gin.ntt.net	Japan
18	392.6634 ms	158.205.192.173	ae0.ostcr01.idc.jp	Japan
19	402.713 ms	158.205.192.86		Japan
20	395.828 ms	158.205.121.250	po2.l321.fk1.eg.idc.jp	Japan
21	399.557 ms	154.34.240.254		Japan
22	388.1666 ms	210.152.135.178		Japan

Como era de esperar, este termino siendo el traceroute mas largo, pasando por un camino sumamente raro. De Argentina a Italia, luego a EE.UU. y finalmente a Japón. Sin embargo, si vemos esto desde un punto de vista económico tiene sentido. El trafico desde América Latina a Japon no debe ser muy alto, por lo que no se justifican los altos costos de hacer un enlace mas directo.

Encontramos nuevamente los enlaces transatlánticos de Sparkle. Entrando a ntt.net nos encontramos con:

Esto nos hace inferir que es una empresa Japonesa de telecomunicaciones.

Cuadro 8: Detección de enlaces intercontinentales: u-tokyo.ac.jp

Id	From	To	Avg. RTT	Simbala	GeoIP
1	181.169.12.1 Argentina, SA fibertel.com.ar	200.89.160.21 Argentina, SA fibertel.com.ar	3.3332 ms	no	no
2	200.89.160.21 Argentina, SA fibertel.com.ar	200.89.165.222 Argentina, SA fibertel.com.ar	-	no	no
3	200.89.165.222 Argentina, SA fibertel.com.ar	195.22.220.102 Italy, EU bai.seabone.net	0.7406 ms	no	yes
4	195.22.220.102 Italy, EU bai.seabone.net	195.22.219.17 Italy, EU spa.seabone.net	27.2478 ms	yes	no
5	195.22.219.17 Italy, EU spa.seabone.net	195.22.219.17 Italy, EU spa.seabone.net	-	no	no
6	195.22.219.17 Italy, EU spa.seabone.net	149.3.181.65 Italy, EU	2.6366 ms	no	no
7	149.3.181.65 Italy, EU	129.250.2.227 United States, NA gin.ntt.net	122.0632 ms	yes	yes
8	129.250.2.227 United States, NA gin.ntt.net	129.250.4.13 United States, NA gin.ntt.net	62.4012 ms	yes	no
9	129.250.4.13 United States, NA gin.ntt.net	129.250.2.54 United States, NA gin.ntt.net	10.7836 ms	no	no
10	129.250.2.54 United States, NA gin.ntt.net	129.250.3.86 United States, NA gin.ntt.net	150.99 ms	yes	no
11	129.250.3.86 United States, NA gin.ntt.net	129.250.6.188 United States, NA gin.ntt.net	15.103 ms	no	no
12	129.250.6.188 United States, NA gin.ntt.net	129.250.2.255 United States, NA gin.ntt.net	-	no	no
13	129.250.2.255 United States, NA gin.ntt.net	61.200.80.218 Japan, AS gin.ntt.net	10.498 ms	no	yes
14	61.200.80.218 Japan, AS gin.ntt.net	158.205.192.173 Japan, AS ostcr01.idc.jp	-	no	no
15	158.205.192.173 Japan, AS ostcr01.idc.jp	158.205.192.86 Japan, AS	10.0496 ms	no	no
16	158.205.192.86 Japan, AS	158.205.121.250 Japan, AS eg.idc.jp	-	no	no
17	158.205.121.250 Japan, AS eg.idc.jp	154.34.240.254 Japan, AS	3.729 ms	no	no
18	154.34.240.254 Japan, AS	210.152.135.178 Japan, AS	-	no	no

El enlace numero 4 es el primer enlace intercontinental entre Argentina e Italia. En este caso el método de Simbala si logra identificarlo, mientras que GeoIP tiene datos incorrectos. El próximo enlace es el numero 7 entre Italia y Estados Unidos. Ambos métodos logran identificarlos. Luego Simbala no clasifica bien los enlaces numero 8 y 10. Solo GeoIP logra identificar satisfactoriamente el enlace numero 13.

2.2. Caching

Este experimento fue una casualidad. Al hacer dos requests a google.com, notamos que mientras una traza hacia un salto intercontinental, la otra no. Allí nos dimos cuenta que en realidad lo que sucedía era que Fibertel hace caching, lo que baja el RTT en un tercio. Suponemos que esto debe ser sumamente útil para proveer a usuarios que utilizan streaming y P2P.

Cuadro 9: traceroute: google.com sin caching

Hop	Avg. RTT	IP Address	Host name	Location
1	10.6688 ms	181.169.12.1	1-12-169-181.fibertel.com.ar	AR, SA
2	* * * * *			
3	* * * * *			
4	* * * * *			
5	20.2096 ms	200.89.160.21	21-160-89-200.fibertel.com.ar	AR, SA
6	14.3278 ms	200.89.165.129	129-165-89-200.fibertel.com.ar	AR, SA
7	12.5566 ms	200.89.165.150	150-165-89-200.fibertel.com.ar	AR, SA
8	* * * * *			
9	10.9052 ms	209.85.251.86	209.85.251.86	US, NA
10	40.759 ms	209.85.252.42	209.85.252.42	US, NA
11	38.5816 ms	216.239.58.221	216.239.58.221	US, NA
12	38.1802 ms	216.58.202.4	gru06s26-in-f4.1e100.net	US, NA

Cuadro 10: traceroute: google.com con caching

Hop	Avg. RTT	IP Address	Host name	Location
1	11.1854 ms	181.169.12.1	1-12-169-181.fibertel.com.ar	AR, SA
2	* * * * *			
3	* * * * *			
4	* * * * *			
5	21.9184 ms	200.89.165.33	33-165-89-200.fibertel.com.ar	AR, SA
6	15.066 ms	200.89.164.26	26-164-89-200.fibertel.com.ar	AR, SA
7	* * * * *			
8	11.6574 ms	181.30.241.187	187-241-30-181.fibertel.com.ar	AR, SA

3. Conclusión

Poder trazar la ruta que un paquete potencialmente puede tomar es sumamente importante, particularmente para podes identificar saltos donde la red se puede congestionar o tener errores. Esta funcionalidad es ampliamente utilizada a día de hoy, si bien en este trabajo practico vimos un método para la implementación de esta herramienta, existen otras maneras de implementarla.

El método propuesto en el TP consiste en abusarse de los paquetes ICMP para poder obtener la ruta del origen al destino, el problema de esto es que es posible que la ruta cambie durante la ejecución obteniendo así una ruta invalida (es una posibilidad minima que puede ser mitigada fácilmente ejecutando varias veces el algoritmo), en varias ocasiones se contemplo incluir la traza de rutas en los protocolos y paquetes, un método fue propuesto en el RFC 791 (*record route*) y el RFC 1393 (*traceroute con IP Option*). El primer método consiste en guardar las direcciones de 32 bits en el header del paquete a medida que transita la red, mientras que el segundo consiste en que los enrutadores sepan que se esta haciendo un traceroute y envíen paquetes identificándose como saltos al origen, el método con *record route* existe aun a día de hoy pero esta limitado a guardar a lo sumo 9 saltos por limitaciones de espacio en el header (creemos que esto se debe a que fue planteado en 1981, antes de la Internet moderna), mientras que el *traceroute con IP Option* requería que todos los enrutadores soporten la opción IP apropiada, así que como era de esperar nunca fue implementado y fue deprecado en 2012 por el RFC 6814.

Respecto a los resultados obtenidos con nuestra herramienta, casi todos se correspondieron con lo que esperábamos. Sin embargo, el de Japón fue el mas interesante, mirando el mapa de enlaces intercontinentales podemos ver que la manera mas directa hacia el destino es a través del océano pacifico, empleando los enlaces que hay sobre la costa oeste de Estados Unidos. Para poder llegar a dichos enlaces hay varias formas, si bien se tomo un camino por Europa, también es posible tomar varios caminos desde Argentina hacia la costa este de Estados Unidos, y luego recorrer hacia la costa oeste de dicho país. Lo mas importante a destacar es que esto deja en claro que la distribución de fibra óptica a lo largo del mundo esta pensada en base al trafico mas que en minimizar la latencia a nivel mundial, pero al igual que dijimos durante el análisis, esto tiene sentido considerando lo reducido que debe ser el trafico de Argentina hacia Japón.

De los dos métodos discutidos, ambos tuvieron sus falencias. Esta claro que el método infalible seria un base de datos como la de GeoIP que este constantemente actualizada, eso permitiría erradicar toda ambigüedad, sin embargo, la naturaleza dinámica y la diversidad burocrática de la Internet hace que sea imposible tener dicha información actualizada, lo cual llevo a los errores vistos. El método de Simbala, por otro lado, es susceptible a la congestión de la red ya que se basa en outliers y en mediciones de tiempo, esto también lo hace sumamente débil ya que si el servidos no responde a los paquetes de *Time Exceeded* el método no se puede aplicar sobre ese salto, sin embargo, cuando la información estaba disponible los resultados fueron aceptables. En la practica, lo prudente seria aplicar varios criterios distintos para evaluar si en un salto hubo un enlace intercontinental, y luego tomar una decisión según la fiabilidad de cada unos de los criterios en cuestión.

Referencias

[1] Carna Botnet. Internet census 2012: Port scanning using insecure embedded devices. http://internetcensus2012.bitbucket.org/paper.html, 2013.

[2] John M Cimbala. Outliers. http://www.mne.psu.edu/cimbala/me345/Lectures/Outliers.pdf, 2011.

[3] RFC 792 (ICMP). http://www.ietf.org/rfc/rfc792.txt.

[4] Martin Erich Jobst. Traceroute anomalies. http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2012-08-1/NET-2012-08-1_02.pdf, 2012.

[5] Traceroute (Wikipedia). http://en.wikipedia.org/wiki/Traceroute.