Traceroute

22 de octubre de 2014

Universidad de Buenos Aires - Departamento de Computación - FCEN

Integrantes:

- Gallardo, Guillermo L.U.: 32/10 gagdiez.c@gmail.com
- Fixman, Martin L.U.: 391/11 martinfixman@gmail.com
- Matayoshi, Leandro L.U.: 79/11 leandro.matayoshi@gmail.com
- Szyrej, Alexander L.U.: 642/11 alexander.szyrej@gmail.com

Índice

1. Introducción

1.1. Objetivos generales

ICMP es el módulo del protocolo TCP/IP que se encarga de proveer mensajes de error y de control cuando se produce alguna anormalidad en el envío de un datagrama IP. Existen distintas herramientas que se apoyan sobre este módulo para obtener información acerca del estado de las rutas que debe atravesar un paquete para llegar al destino, como por ejemplo, el valor del RTT (round-trip-time) entre 2 hosts. Entre estas herramientas se destacan los comandos *ping* y *traceroute*.

El objetivo de este trabájo práctico es realizar nuestra propia implementación de *traceroute* y utilizar la misma para comprender el funcionamiento de las comunicaciones a larga distancia sobre internet. Para esto último realizaremos un análisis de la distribución geográfica del conjunto de enlaces y routers atravesados por los paquetes ICMP. A su vez intentaremos detectar cuando dichos paquetes atraviesan enlaces intercontinentals utilizando solamente los valores de los RTT.

Por todos estos motivos, los hosts destino de los paquetes son cuatro universidades ubicadas en otros continentes, las mismas son:

- **Oxford** University of Oxford, Londres, Reino Unido
- **MSU** Московский государственный университет (Universidad Estatal de Moscú), Moscú, Federación Rusa
- **Queensland** University of Queensland, Queensland, Australia
- **Tsinghua** 清华大学 (Universidad de Tsinghua), Beijing, República Popular China

1.2. Tipos de mensajes ICMP

Los mensajes de ICMP se transmiten en forma de datagramas. El emisor puede ser tanto un host como router, y el destino es siempre la dirección source del datagrama IP que motivó el mensaje.

Entre los tipos de mensajes de error más comunes se destacan:

- Echo reply: Respuesta a echo request. Los datos recibidos por el request deben ser incluídos en el mensaje. type: 0
- Destination host unreachable: El router no encuentra en su tabla una dirección a la cual forwardear el paquete. type: 3
- Redirect: El router que recibe el paquete detecta que otro router ofrece un camino más efectivo para forwardear el paquete. type: 5
- Echo request: Se espera que los datos enviados sean recibidos nuevamente en un mensaje echo reply. type: 8
- Time exceeded: Mensaje generado por un router para indicarle al host emisor de un datagrama que su TTL ("Time to live") ha alcanzado el valor 0.

1.3. Ping

El comando ping es una tool que permite testear la actividad de un host dentro del protocolo IP y medir el RTT entre el dispositivo que ha ejecutado el comando y el host de interés. El mismo se basa en la emisión de mensajes *echo request*, y sus respectivas respuestas *echo reply*. El RTT queda determinado entonces por el tiempo que tarda el host emisor en recibir la respuesta. Los requests para los cuales no se recibe ninguna respuesta son registrados como paquetes perdidos.

1.4. Traceroute

A diferencia de ping, *traceroute* provee información acerca de la ruta que siguen los datagramas para arribar al destino. Esta herramienta arma un registro ordenado con los routers que ha atravesado el paquete, junto con el valor de los RTT para cada uno de ellos. En la siguiente sección explicaremos el mecanismo de esta herramienta con mayor detalle.

2. Desarrollo

2.1. Implementando traceroute

Como dijimos anteriormente, *traceroute* es una herramienta de dignóstico que se utiliza para hacer un análisis del estado de la conexión entre 2 hosts comunicados a través del protocolo TCP/IP. Este comando pone especial énfasis en los distintos tramos (determinados por los hops entre routers) que va debe atravesar el paquete para llegar a destino. Mediante este tipo de análisis es posible obtener información de gran interés, como por ejemplo identificar los enlaces con mayor RTT (determinantes en el cálculo del RTT "global", entre el host origen y el destino). De esta manera es posible caracterizar los distintos enlaces: RTT's altos pueden corresponderse a enlaces submarinos o terrestres de largas distancias, o enlaces con poca velocidad de transmisión.

La idea básica del algoritmo consiste en enviar paquetes con TTL's incrementales hacia el nodo destino y obtener información en función de los datagramas ICMP obtenidos, tal como muestra el siguiente diagrama:

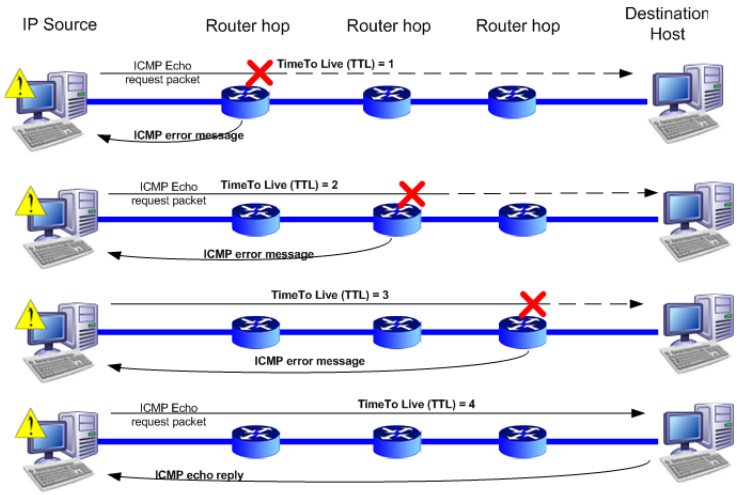

Figura 1: Mecanismo de acción de traceroute

Cada vez que un paquete es forwardeado hacia un router, el valor de su TTL se decrementa en una unidad. Cuando un routers advierte que el TTL de un paquete ha alcanzado el valor 0, envía un mensaje ICMP hacia el host fuente del mensaje, de tipo "time exceeded". Conociendo la hora exacta de emisión del paquete original y hora en la cual arriba el mensaje de error es posible determinar el RTT entre el host en el cual se corre el algoritmo y el router generador del paquete ICMP. El algoritmo se extiende a cada uno de los nodos intermedios: En la primera iteración se envía un paquete con valor de TTL 1. Por lo tanto el router a 1 hop de distancia envía un mensaje de error, y el algoritmo puede calcular el RTT entre el origen y dicho router. Generalizando, en la iteración i se envía un paquete con valor de TTL i, y utilizando la información recibida, el algoritmo puede calcular el RTT entre el origen y el router ubicado a i hops. El algoritmo finaliza o bien cuando se recibe una respuesta afirmativa por parte del destino; o bien cuando se alcanza el límite de TTL permitido.

2.1.1. Algunas consideraciones y detalles implementativos

Para implementar el comando utilizamos la librería *Scapy* de Python.

Traceroute por default envía una secuencia de paquetes UDP hacia el nodo destino. Paquetes del estilo TCP SYN también pueden ser usados. En nuestra implementación la alternativa utilizada son los paquetes ICMP de tipo "echo request" . Los mensajes recibidos serán por lo tanto del tipo "time exceeded" para los nodos intermedios y "echo reply" cuando el paquete llega a destino.

La función debe recibir como argumentos el valor de TTL inicial con el que envía el primer paquete y el máximo de saltos permitidos hasta alcanzar el destino. En nuestro caso utilizamos los valores por defecto: 1 y 30 respectivamente.

Otro de los parámetros que debe especificarse es el número de reintentos que debe realizarse por cada valor de TTL. ICMP, al igual que IP, es una capa "sin garantías" en el envío de mensajes. Los routers hacen el "mejor esfuerzo" para lograr el envío de los paquetes, pero no hay garantías de que los mismos alcancen el destino. Teniendo esto en cuenta, es posible que los mensajes ICMP enviados por los routers no lleguen

al emisor. En dicho caso, es deseable que se realice más de un intento para un determinado valor de TTL. Usualmente la mayoría de las implmentaciones utilizan 2 o 3 como valor. En nuestro caso decidimos realizar 10 intentos por cada TTL. Este valor nos protege contra otro aspecto sensible del algoritmo, descripto a continuación.

Un paquete no llega al mismo destino atravesando siempre las mismas rutas. Los nodos intermedios pueden caerse o levantarse. Determinados enlaces pueden congestionarse o liberarse en distintos momentos. Supongamos que el algoritmo envía hacia el destino un paquete con TTL 10. ¿Qué garantías tenemos de que, en caso de ser necesario un segundo intento porque no ha llegado ningún mensaje de error, el router alcanzado luego del décimo hop sea el mismo? Ninguna. En otras palabras, dependiendo la ruta que siga el datagrama, el router alcanzado luego del décimo hop será uno u otro. Para tener cierto control respecto a esta situación, por cada valor de TTL se envían 10 paquetes y se guardan todas las direcciones de IP que han respondido los mensajes de error. Luego, el algoritmo selecciona aquella dirección que ha respondido más veces.

Finalmente, el valor de RTT entre el host que corre el *traceroute* y el router a i hops de distancia es calculado como el promedio de los valores obtenidos por cada intento.

2.2. Estimando Enlaces Intercontinentals

Recordando algunas fórmulas pertenecientes a la capa de enlace, sabemos que el delay está determinado por la siguiente ecuación:

$delay[seg] = T_{tx} + T_{prop}$, donde T_{tx} es el tiempo de transmisión determinado principalmente por la capacidad del canal y la relación señal ruido dentro del mismo, y el tiempo de propagación está determinado por:

$T_{prop} = \frac{D}{V}$, con D la distancia del enlace y V la velocidad de propagación de la forma de onda en el medio físico.

El RTT equivale a $2 * delay$.

Si analizamos exclusivamente la capa de enlace punto a punto del modelo de capas de Internet, podemos llegar a la conclusión de que el factor de mayor peso en el cálculo del RTT es la distancia del enlace. Dado que se envía un único datagrama con una cantidad despreciable de datos de test (caracteres ASCII pertenecientes al "echo request"), el tiempo de transmisión del mismo dentro del enlace es ínfimo (independientemente de la velocidad de transmisión de dicho enlace). Por lo tanto, desde este punto de vista es lógico pensar que los resultados con mayor valor de RTT corresponderán a enlaces submarinos. Suponemos que el tiempo que le toma a un paquete ir de un nodo a otro por tierra es significativamente menor al que le toma atravesar un enlace intercontinental, ya que estos últimos cubren distancias mucho mayores.

Sin embargo, la capa de nivel 1 no es la única que debemos tener en cuenta a la hora de realizar el análisis. Podría suceder que por algún motivo (congestión, por ejemplo), el paquete sea almacenado temporalmente en un buffer antes de ser forwardeado hacia el próximo router. O que algún conflicto de índole similar suceda a nivel Ethernet. Por lo tanto, estos factores pueden influir en un resultado con RTT alto sin que el router involucrado provenga necesariamente de un enlace submarino.

El *zscore* es una medida que nos permite medir el peso de un hop respecto a los demás. (En particular, respecto a la media de todos los hops de la ruta, y considerando también el desvío standard de los datos). La fórmula utilizada para calcularlo es:

$ZRTT_i = \frac{RTT_i - RTT_{mean}}{SRTT}$

$ZRTT_i$ representa el RTT para el i-ésimo hop. Para calcularlo, obtenemos la distancia en tiempo que hay entre los pares de nodos vecinos y asumimos que los mismos siguen una distribución normal. Finalmente fijamos un umbral de Zscore para determinar los tramos son submarinos.

3. Resultados y análisis

Luego de correr todos los experimentos notamos que la dirección IP de los nodos no necesariamente refleja su posición geográfica real; esta es necesaria para saber cuales ejes son intercontinentales, y por ende fijar un umbral $\boldsymbol{\mu}$ válido.

Intentamos hacer uso de la herramienta www.geoiptool.com para orientarnos en el curso de las rutas y algunas veces encontrábamos que no tenían sentido las ubicaciones que nos proponía esta herramienta.
Un ejemplo de esto mismo seria:

Host name	***IP***	***RTT(media) ms***
ET6-0-0-0-GRTBUECU1.red.telefonica-wholesale.net	94.142.103.153	30.104

Dicha IP figura en geoiptool como localizada en España. Es decir que de Buenos Aires, la ruta pasará por España antes de llegar a Estados Unidos, todo con un RTT medio de 30ms. Esto no parece para nada razonable, pero el nombre del host nos presenta información que puede contrastar aquella proveída por geoiptool. GRT**BUE**CU1 indicaría que la IP realmente pertenece a Buenos Aires, algo mucho más lógico.
Afortunadamente capturamos los nombres de los hosts en cada salto y la mayoria tiene indicios de ubicación en sus nombres. Claro que no siempre encontramos hosts 'bien nombrados'.

Otro ejemplo podría ser *be2384.ccr21.**lpl**01.atlas.cogentco.com* con el código IATA de Liverpool.

En las siguientes tablas se muestra la ubicación (no necesariamente aunque muy probablemente correcta) del host en el que termina cada *hop*, su IP, la media estadística del Round Trip Time a ese hop, y el Z-score de este host tomando como valor la diferencia de tiempo con el host anterior. Como un paquete con TTL igual a 0 terminaría en el cliente en un tiempo infínitisamente chico, se toma el primer "host" con distancia 0.
Para simplificar las tablas y mejorar la comprensión, en las tablas siguientes las filas correspondientes a los links cuyos hosts están en diferentes continentes (visto "a ojo") están destacados. También el link no-intercontinental con Z-score más grande está marcado, ya que va a ser análizado luego.

3.1. Moscow State University (Rusia)

3.1.1. Ruta

Ubicación	*IP*	*RTT(media) ms*	*ZScore*
Buenos Aires	200.51.240.181	31.018	0.220
Buenos Aires (telefónica)	94.142.103.153	30.104	-0.265
Miami (telefónica)	94.142.123.22	164.933	1.795
Dallas (telefónica)	94.142.127.105	199.813	0.278
Dallas/Ft Worth (cogentco)	154.54.13.225	221.383	0.077
Dallas/Ft Worth (cogentco)	154.54.7.45	212.266	-0.389
Kansas (cogentco)	154.54.2.113	214.954	-0.210
Chicago (cogentco)	154.54.6.86	215.663	-0.240
Toronto (cogentco)	154.54.27.182	228.466	-0.056
Montreal (cogentco)	154.54.30.206	226.107	-0.286
Liverpool (cogentco)	154.54.44.138	294.383	0.785
Amsterdam (cogentco)	154.54.77.245	307.648	-0.049
Hamburgo (cogentco)	154.54.74.122	264.320	-0.908
Estocolmo (cogentco)	154.54.63.2	327.827	0.713
Helsinki (cogentco)	154.54.62.250	332.424	-0.181
Moscú	149.6.58.42	333.589	-0.233
Moscú (runnet)	194.85.40.229	525.315	2.658
Moscú (runnet)	194.190.254.118	357.420	-2.798
Moscú (runnet)	93.180.0.172	344.003	-0.454
Moscú	188.44.33.1	346.421	-0.214
Moscú	188.44.50.103	347.007	-0.242

Las ciudades fueron atravesadas en el siguiente orden:
Buenos Aires → Miami → Dallas → Kansas → Chicago → Toronto → Montreal → Liverpool → Amsterdam → Hamburg → Stockolm → Helsinki → Moscow

Como se puede ver, la conexión primero pasa por varios hosts en los Estados Unidos y en Canadá antes de pasar por un cable transátlantico entre Montreal y Liverpool. A pesar de que estos dos son por lejos los links más largos, el link con Z-score más alto es el que entra a la red en Moscú. Esto se puede deber a que los routers que administran internet dan mucha menos prioridad a paquetes ICMP que a paquetes que solamente tienen que forwardear a su red. Por esta razón, se puede ignorar sin consecuencias graves y tomar al host siguiente con Z-score más grande.

Figura 2: Mapa de la ruta atravesada para llegar a la universidad de Rusia

3.2. Tsinghua

Para el caso de China, la ruta comienza por Estados Unidos como el caso anterior, pero en lugar de cruzar a Europa por el Este, los paquetes viajan por el pacífico. Veamos la siguiente tabla.

Ubicación	***IP***	***RTT(media) ms***	***ZScore***
Buenos Aires	(200.51.240.156)	31.907	0.276
Buenos Aires	(84.16.9.233)	30.095	-0.467
Miami	(94.142.121.222)	165.469	2.558
Miami	(94.142.122.249)	165.140	-0.434
Miami	(84.16.12.238)	167.934	-0.365
Miami	(63.243.152.45)	164.176	-0.510
Ashburn	(66.198.154.177)	215.945	0.714
Dallas	(66.198.154.118)	254.661	0.427
Dallas	(66.110.56.6)	267.287	-0.149
Los Angeles	(66.110.57.82)	256.103	-0.674
Los Angeles	(66.110.59.182)	257.170	-0.403
Chongming	(101.4.117.213)	428.694	3.355
Chongming	(101.4.117.97)	428.055	-0.441
Beijing	(101.4.116.146)	424.990	-0.495
Beijing	(101.4.118.78)	427.001	-0.383
Beijing	(202.112.38.10)	425.663	-0.456
Beijing	(118.229.4.66)	425.672	-0.427
Beijing	(118.229.4.34)	438.217	-0.150
Beijing	(118.229.2.74)	430.018	-0.608
Beijing	(118.229.2.69)	425.922	-0.517
Beijing	(118.229.8.6)	427.147	-0.400
Beijing (Tsinghua)	(166.111.4.100)	426.084	-0.450

Las ciudades fueron atravesadas en el siguiente orden:
Buenos Aires → Miami → Ashburn → Dallas → Los Ángeles → Chongming → Beijing

El link continental más largo, en Ashburn, tiene un Z-score de 0.714. Aunque es claramente un outlier, lo podemos usar para la medición de $\boldsymbol{\mu}$. El link intercontinental más corto tiene un Z-score de 2.558.

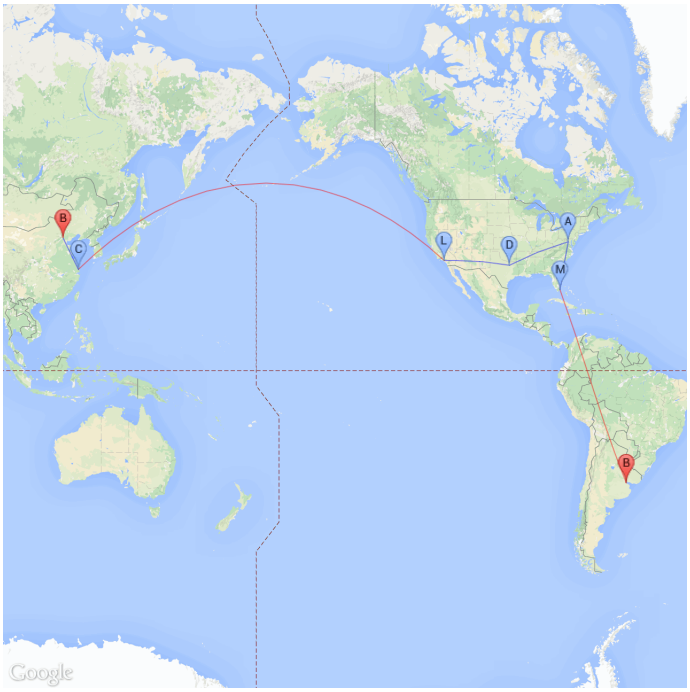

Figura 3: Mapa de la ruta atravesada para llegar a la universidad de China

3.3. Oxford

Ubicación	*IP*	*RTT(media) ms*	*ZScore*
Buenos Aires	(200.51.240.181)	37.154	0.454
Buenos Aires	(84.16.9.233)	51.008	-0.010
Miami	(5.53.5.138)	210.915	2.899
Miami	(94.142.123.5)	174.734	-1.006
Miami	(63.243.152.45)	175.240	-0.276
Ashburn	(66.198.154.177)	210.857	0.424
Ashburn	(216.6.87.1)	286.600	1.223
Newark	(216.6.87.138)	277.667	-0.464
Newark	(66.198.70.1)	307.315	0.305
Londres	(66.198.70.26)	403.133	1.622
Londres	(80.231.130.42)	298.757	-2.365
Londres	(195.219.100.82)	285.687	-0.546
Londres	(146.97.33.2)	289.063	-0.219
Londres	(146.97.37.206)	291.484	-0.238
Londres	(193.63.108.129)	281.783	-0.479
Londres	(193.63.108.134)	294.226	-0.038
Londres	(193.63.109.114)	289.814	-0.374
Londres	(192.76.21.21)	280.879	-0.464
Londres	(192.76.22.201)	282.251	-0.259
Londres	(192.76.32.66)	286.466	-0.202
Londres (Oxford)	(129.67.242.155)	301.388	0.011

Las ciudades fueron atravesadas en el siguiente orden:
Buenos Aires → Miami → Ashburn → Newark → London → Oxford

En este caso el link a Ashburn también es el que tiene el Z-score más alto entre los links continentales con 1.223, mientras que el link intercontinental más bajo es de 1.622.

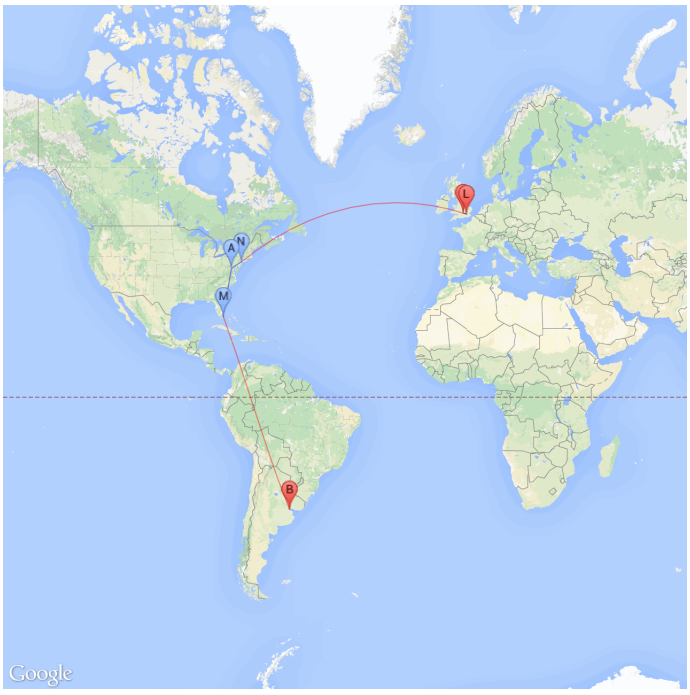

Figura 4: Mapa de la ruta atravesada para llegar a la universidad de Inglaterra

3.4. Queensland

Ubicación	*IP*	*RTT(media) ms*	*ZScore*
Buenos Aires	(200.51.240.156)	31.896	0.140
Buenos Aires	(84.16.9.233)	30.748	-0.509
Miami	(94.142.123.14)	165.811	2.170
??? (213.140.49.13)	(213.140.49.13)	230.069	0.776
Palo Alto	(64.125.13.113)	229.511	-0.497
Sydney	(208.185.52.74)	392.486	2.720
Sydney	(202.158.194.176)	391.129	-0.513
Sydney	(113.197.15.57)	419.913	0.079
Brisbane	(202.158.194.54)	407.555	-0.729
Brisbane	(202.158.194.213)	394.376	-0.745
Queensland	(202.158.209.3)	394.951	-0.475
Queensland	(113.197.8.34)	391.479	-0.555
Queensland	(130.102.159.1)	392.041	-0.475
Queensland	(130.102.0.242)	401.994	-0.291
Queensland	(130.102.82.28)	403.280	-0.461
Queensland	(130.102.131.70)	396.128	-0.627

Las ciudades fueron atravesadas en el siguiente orden: Buenos Aires → Miami → Hermosa Beach → Sydney → Brisbane → Queensland

Aquí el link continental con mayor Z-score tiene uno de 2.720, mientras que el link continental con mayor Z-score es de 0.776.

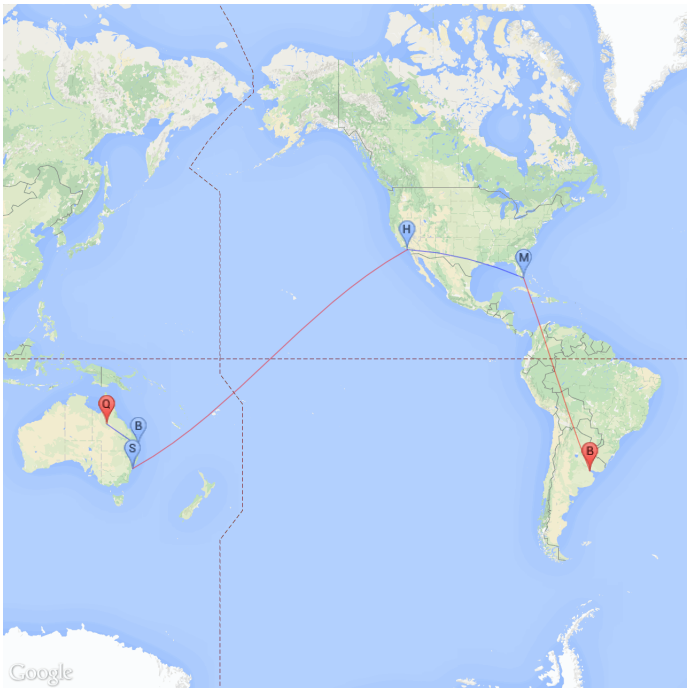

Figura 5: Mapa de la ruta atravesada para llegar a la universidad de Australia

Entre todos los hosts, el link intercontinental con Z-score más bajo es el de Montreal a Liverpool, con $Z_{rtt} = 0{,}785$, mientras que el link intracontinenteal con Z-score más alto es uno de un host a otro en la ciudad as Ashburn, con $Z_{rtt} = 1{,}233$.

4. Conclusiones

Ya que el link intercontinental con menor Z_{rtt} tiene uno más chico que el link intracontinental que tiene este valor mayor, podemos ver que es imposible definiar un $\boldsymbol{\mu}$ que clasifique estos links de esta manera con 100 % de presición. Sin embargo, se puede ver que ambos valores son outliers, ya que todos los otros links intracontinentales son menor que 1 y todos los otros links intercontinentales son mayores que estr número.

Con estos valores, podemos definir $\boldsymbol{\mu = 1}$ como resultado, y esto va a poder clasificar los links de una manera correcta para todos los links menos los dos extremos. Por otro lado, si se elige $\boldsymbol{\mu} = 0{,}780$ ó $\boldsymbol{\mu} = 1{,}5$ uno de los dos extremos va a tener un valor válido (y solo un valor va a ser diferente en el "mundo real" que en este análisis), pero como los dos valores son outliers, el primer caso va a tener más proabilidad de dar resultados correctos si se le agregan más puntos cerca del extremo.

5. Referencias

- **Computer Networks: A Systems Approach**, *Larry L. Peterson and Bruce S. Davie.*
- **Computer Networks**, *Andrew S. Tanenbaum*
- **http://submarinecablemap.com/**