

# Trabajo Práctico N° 2: Rutas en Internet

Leandro Ezequiel Barrios, Gonzalo Benegas, Martin Caravario, Pedro Rodriguez

## Resumen —

En el presente Trabajo Práctico nos proponemos experimentar con herramientas y técnicas frecuentemente utilizadas a nivel de red. En particular, nos centraremos en la utilización de `traceroute` como herramienta para medir tiempos y RTTs.

## I. INTRODUCCIÓN

Los objetivos del presente Trabajo Práctico son múltiples: por un lado, entender los protocolos involucrados en cada comunicación o intercambio de datos entre un usuario y un servidor, estando ambos conectados a Internet. Vimos en clase que estos protocolos son tanto de nivel de capa de transporte (TCP, UDP) como de aplicación (DNS, SMTP). Por otro lado, para afianzar nuestros conocimientos, implementaremos una herramienta que cumpla la misma función que `traceroute` por nuestra cuenta. La función de esta herramienta es la de detectar, a partir del nombre de un dominio (host), cada uno de los nodos intermedios por los que pasa el paquete antes de llegar a la dirección IP correspondiente a dicho host. En este TP, analizaremos los RTT de paquetes enviados a distintas universidades del mundo.

## II. CARACTERIZANDO RUTAS

### A. Implementación de tool para hacer `traceroute`

Implementamos la herramienta `traceroute2` en python, usando la librería `scapy`. Para esto, armamos varios paquetes echo-request, iniciándolos con un TTL entre 1 y alguna cantidad de hops variable, que nos permita llegar al destino (en general, 30 o 40 hops son más que suficientes). A cada paquete ICMP que es respondido, le corresponderá un RTT diferente y, los paquetes que no son respondidos después de un tiempo de timeout determinado, asumimos que fueron recibidos y no respondidos o que se perdieron en la red, marcando que el RTT y resto de la información correspondiente a dicho paquete con un \*.

Una herramienta extra que implementamos para facilitar la experimentación fue una función que llamamos `DNSInverso`, que para cada router nos permite reconocer su nombre. De esta forma podemos detectar, para cada máquina de la que obtenemos respuesta en algún hop, a qué organización pertenece (si es que pertenece a alguna). Y, en particular, si pertenece a la universidad a la que estamos haciendo el `traceroute`.

A partir de los valores de RTT obtenidos para cada nodo intermedio de la ruta rastreada por `traceroute2`, graficamos el RTT observado entre dicho nodo y el nodo origen.

### B. Adaptación de la tool

Después de haber experimentado con la tool original, enviando paquetes echo-request a varias universidades alrededor del mundo, adaptamos dicha tool para que, una vez

terminada la búsqueda, calcule el *valor standard* o *valor Z* ( $ZRTT$ ) en cada salto con respecto a los saltos realizados en la ruta global de la siguiente manera: si  $RTT_i$  es el RTT medido para el salto  $i$ , se define

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

donde  $\overline{RTT}$  y  $SRTT$  son el promedio y el desvío estándar de una lista que contiene las diferencias de RTT entre cada hop y el hop que lo antecede en la ruta que realiza el paquete.

Luego, para cada hop  $i$  tendremos un valor  $|ZRTT_i|$ . Por cómo definimos a  $ZRTT_i$ , se observa que a mayor *valor Z* para un hop, mayor será la variación del tiempo que se tarda en atravesar dicho hop en comparación con el promedio del tiempo que se tarda en atravesar cada hop de la ruta.

A partir de esta adaptación esperamos introducir en los gráficos de  $RTT = f(\text{número de hop})$  que habíamos hecho con la tool original la nueva información obtenida sobre cada hop. Esperamos que esto nos permita detectar aquellos hops que corresponden a enlaces submarinos. En particular, si el hop  $i$  tiene un  $|ZRTT_i|$  mucho mayor que el resto, sería un indicador de que en el hop  $i$  el paquete atraviesa un enlace submarino.

## III. EXPERIMENTACIÓN: TRACEROUTE A UNIVERSIDADES DEL MUNDO

A partir de la herramienta `traceroute2` y tomando como IP origen la de la PC de uno de los integrantes del grupo, realizamos rastreos para distintas universidades alrededor del mundo. Cada una de estas universidades las identificamos a partir de su URL correspondiente. En el proceso de la experimentación, introducimos los datos obtenidos para cada universidad en un graficador, que dada la lista de direcciones IP recorridas, las localiza geográficamente en un mapa del planeta Tierra, y las une mediante una línea.

Las universidades que terminamos eligiendo fueron: en Noruega (Oslo) a «www.uio.no», en Japón (Tokyo) a «www.u-tokyo.ac.jp», en Tanzania (Dar es Salaam) a «www.udsm.ac.tz» y en Nueva Zelanda (Auckland) a «www.auckland.ac.nz».

De esta forma, intentamos expresar de la forma más gráfica posible el camino a escala global que debió efectuar cada paquete para llegar desde nuestra PC hasta algún servidor de dicha universidad, ubicada a decenas de miles de kilómetros de nosotros.

Uno de los problemas con que nos encontramos encontrar universidades para las cuáles los trayectos efectuados por los paquetes tuvieran distintas características topológicas (por ejemplo que los paquetes atravesaran enlaces submarinos o que las universidades queden en distintos continentes), a fin de poder realizar posteriormente un análisis más

completo e interesante de los RTTs y ZRTTs correspondientes a cada trayectoria.

Dos de los problemas con los que nos encontramos durante la experimentación fueron:

- Tuvimos problemas con el envío de paquetes a través de conexión wi-fi. Es por esto que tuvimos que realizar todas las mediciones con una misma computadora, conectada a internet a través de una conexión cableada.
- Encontramos muchas universidades, como por ejemplo la de Nairobi ([www.uombi.ac.ke](http://www.uombi.ac.ke)), para las cuáles el camino graficado en el mapa terminaba en EEUU, en lugar de de en África, como esperábamos. Para estos casos, supusimos que el problema no era de un bug en nuestra implementación, sino en que dicha página web estaba hosteada en un servidor localizado en EEUU. Una posible razón por la cuál dicha universidad querría hacer esto podría ser, por ejemplo, que el costo de hostear un servidor en Kenia es mayor que en EEUU.
- En todos los casos analizados, hubieron muchos nodos intermedios que no contestaban nuestro *ICMP Request*, con lo cuál habían muchos *hops* que no éramos capaces de localizar en el mapa, lo cuál restaba precisión a nuestra representación de la trayectoria de los paquetes.
- Hubieron algunas universidades para las cuáles (por ejemplo, al enviar 40 paquetes a [www.udsm.ac.tz](http://www.udsm.ac.tz)) había un primer router localizado en dicha universidad nos contestaba, pero después de obtener esta respuesta, dejábamos de obtener respuesta. De esta forma, no logramos obtener una respuesta de la IP destino. Creemos que este último router que nos contestaba estaba actuando como firewall de la red interna de dicha universidad.

## A. Gráficos y análisis

### A.1 Universidad de Auckland

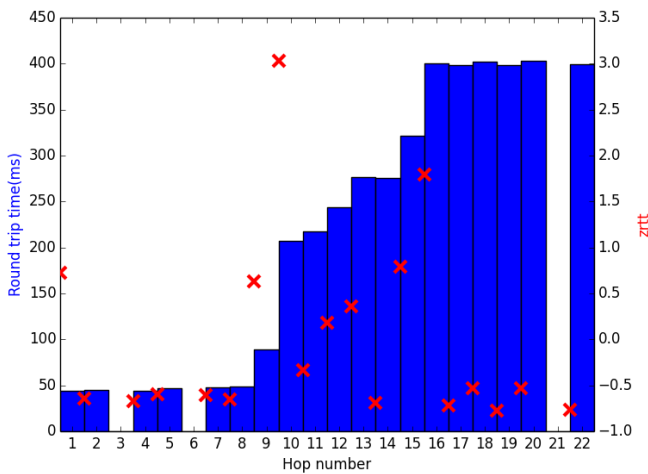


Figura 1. Auckland

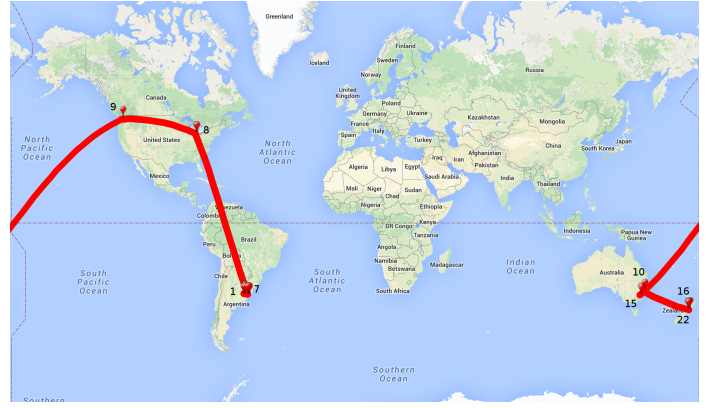


Figura 2. Auckland - mapa

### A.2 Universidad de Tanzania

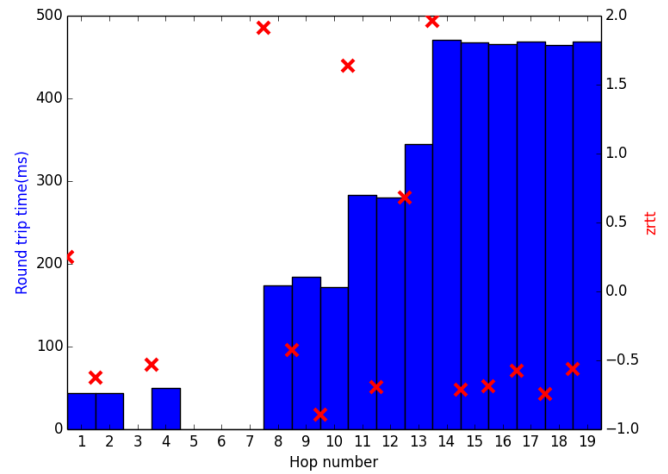


Figura 3. Tanzania



Figura 4. Tanzania - mapa



Figura 6. Tokyo - mapa

A.3 Universidad de Tokyo

A.4 Universidad de Oslo

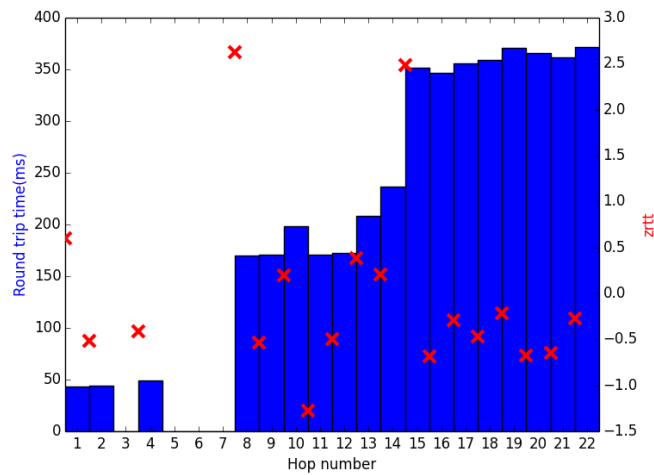


Figura 5. Tokyo

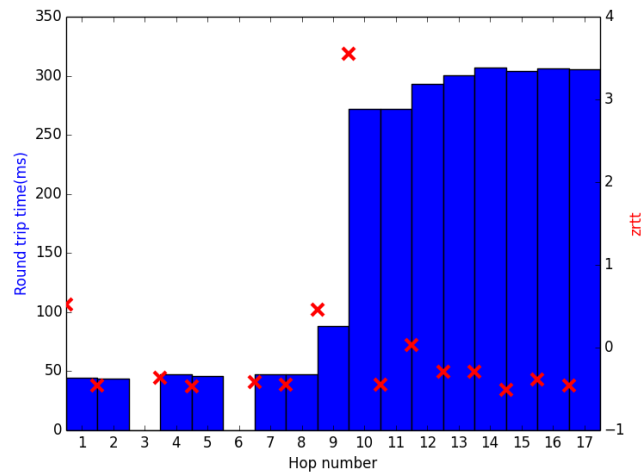


Figura 7. Noruega

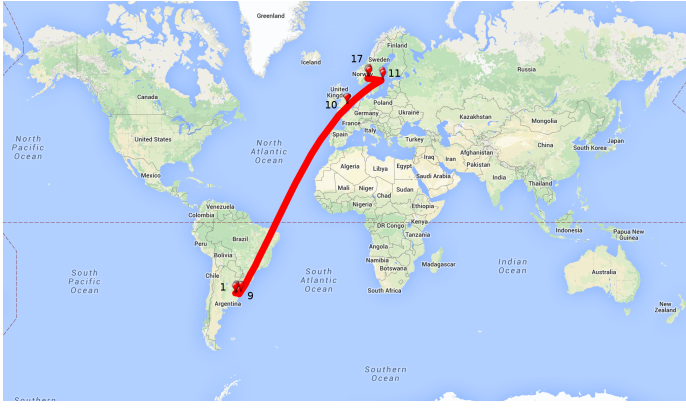


Figura 8. Noruega (Oslo) - mapa

### A.5 Análisis general

Lo primero que nos llamó la atención de estos histogramas fue que los RTTs observados entre dos hops consecutivos no siempre son crecientes. Esto en principio no tiene sentido, pues cualquier paquete, para llegar al hop  $i$ , necesariamente debe pasar antes por el hop  $i - 1$ . Luego, una posible explicación para esto es que alguno/s router/s de la ruta estén utilizando colas de prioridad para los paquetes, y que en particular los ICMP (los que nosotros estamos enviando) tengan una prioridad baja.

### IV. CONTRASTANDO CON LA REALIDAD

A continuación simulamos una conexión mediante sucesivos *pings* (*EchoRequest* + *EchoReply*) a cada una de las universidades analizadas, y estimamos el *RTT* a dicho destino en base a la siguiente ecuación:

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

Con la introducción de este parámetro *alpha*, lo que haremos es ir actualizando el *EstimatedRTT* en cada hop del recorrido del paquete en función del RTT que hubo en el hop anterior y en el actual, dándoles un peso (importancia) a cada uno.

Variamos la cantidad de paquetes enviados ( $n$ ) y *alpha* para obtener los parámetros más adecuados para analizar los resultados y contrastarlos con los resultados obtenidos en la sección anterior. Esta elección previa de parámetros se realizó designando como host a la Universidad de Auc-land.

Es interesante el valor estimado del RTT usando distintos valores de  $n$ . Dada la fórmula de estimación, el tener más mediciones debería moderar el efecto de outliers, resultando en una estimación más uniforme. En efecto, en **Figura ??** se puede observar una variabilidad grande entre haber medido 10, 50 y 100 paquetes, pero un resultado mucho más estable tras haber medido 500 y 1000 paquetes.

Cuando *alpha* está cercano a 0, estamos estimando el RTT hasta el host solo en función de la última medición. A medida que *alpha* se acerque a 1, le estamos dando más importancia a nuestro conocimiento previo sobre la congestión en la red y el valor del RTT. Parece sensato elegir

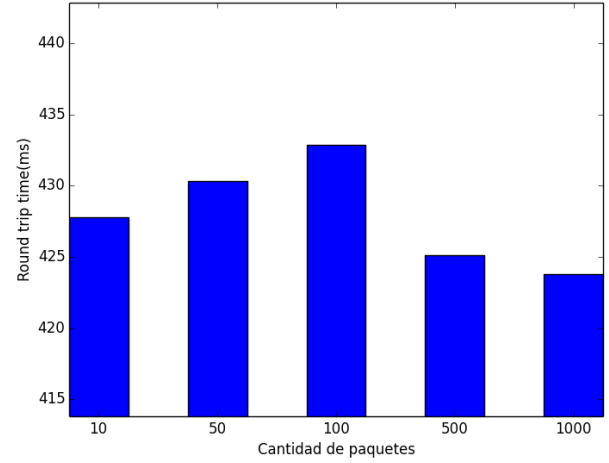


Figura 9. RTT estimado para distintos  $n$ , con  $\alpha=0.8$

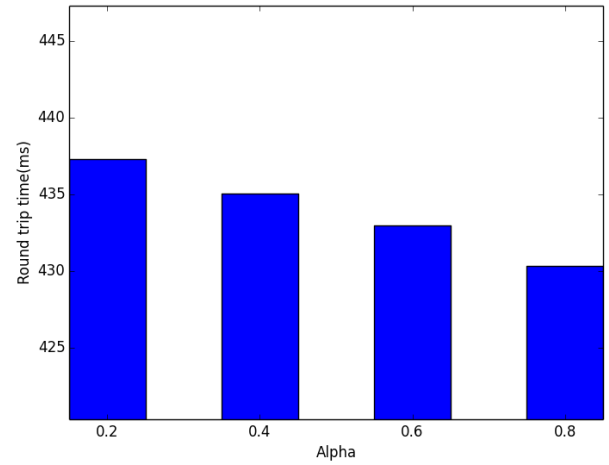


Figura 10. RTT estimado para distintos  $\alpha$ , con  $n=1000$

un *alpha* bastante grande para aprovechar la información previa y hacer una estimación estadística. Un *alpha* chico resultaría en una estimación del RTT muy volátil, afectada por la presencia de *outliers*. De hecho, el estandar TCP define  $\alpha = 0,875^1$ . Nuestros resultados, expuestos en **Figura 10**, muestran un RTT estimado más bajo para los valores de *alpha* más altos. Sin embargo, la diferencia no parece demasiado grande en relación al RTT. Se estima que las diferencias se apreciarían más en redes muy congestionadas.

Por último, se nos pide que analicemos el eventual *throughput* punta a punta que se deriva de la *ecuación de Mathis*:

$$Throughput \leq \frac{MSS}{RTT * \sqrt{p}}$$

donde  $p$  es la probabilidad estimada de pérdida de pa-

<sup>1</sup> <http://tools.ietf.org/html/rfc6298>

quetes, definida de la siguiente manera:

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

$MSS$  es el *MaximumWindowSize*,

Universidad	RTT inicial	Estimated RTT	$p$	Throughput de Mathis
Auckland	399.84	423.81	0.005	48.72
Oslo	305.48	337.61	0.003	78.95
Tanzania	468.56	495.57	0.004	46.58
Tokyo	372.04	394.87	0.006	47.73