



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Práctico 1

## Rutas en Internet

Teoría de las Comunicaciones  
Primer Cuatrimestre de 2017

Integrante	LU	Correo electrónico
Borgna, Agustín	079/15	aborgna@dc.uba.ar
Gonzalez Benitez, Juan	324/14	gonzalezjuan.ab@gmail.com
Lancioni, Gian Franco	234/15	gianflancioni@gmail.com
Vazquez, Cristian	056/10	cristianvazquez4@gmail.com



Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón 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

<b>1. Introducción</b>	<b>3</b>
1.1. ICMP Traceroute . . . . .	3
1.2. Métodos . . . . .	3
1.2.1. Primer ejercicio . . . . .	3
1.2.2. Segundo ejercicio . . . . .	3
<b>2. Mediciones realizadas</b>	<b>4</b>
<b>3. Conclusiones Generales</b>	<b>5</b>

El objetivo de este TP se centra en implementar y usar distintas técnicas y herramientas de diagnóstico a nivel capa de red, en particular las rutas de red que siguen determinados paquetes y los delays respectivos a cada nodo. Esto significa comprender los protocolos necesarios, implementar dichas técnicas, experimentar con ellas y por último estudiar los resultados en un marco analítico que nos permita extraer conclusiones de interés.

## 1. Introducción

### 1.1. ICMP Traceroute

Particularmente, estaremos haciendo uso de interacciones de control (es decir, no se intercambian datos) entre dispositivos de la red que nos permitirán armar una ruta especulativa desde un host a otro. Dichas interacciones pertenecen al *Internet Control Message Protocol (ICMP)*.

El header de dicho protocolo contiene un *tipo* y un *subtipo*. De dichos tipos nos interesan puntualmente 3 tipos de mensaje. Uno de ellos de *request* conocido como '*Echo request*' (tipo 8) que se encarga de enviar a un destino en particular un pedido de '*ping*', es decir que el destinatario rebote el mensaje al remitente.

Luego nos interesan dos tipos de mensaje de *response*, la primera es el '*Echo reply*' que emite el destinatario del '*Echo request*' si el paquete llegó en estado pertinente, y la segunda un '*Time exceeded*' que sucede cuando el *TTL (time to live)* del paquete IP emitido llegó a 0 camino al destinatario (el paquete lo envía cualquier nodo intermedio, en caso de que esté configurado para responder antes dichos escenarios).

Este último tipo de mensajes nos permitirá, a partir de iteraciones incrementales sobre el *TTL* en paquetes del tipo '*Echo request*', armar una lista de IPs de destinos intermedios que respondieron *Time exceeded* antes de obtener un '*Echo reply*'. Este tipo de herramienta se conoce como un **traceroute**<sup>1</sup>, aunque no necesariamente siempre se implementa mediante *ICMP* sino que existen variaciones sobre *TCP SYN*, *UDP* y, sin necesidad de usar *TTLs*, existe una opción sobre IP establecida por *RFC 1393* que permite enviar la mitad de paquetes necesarios para una implementación basada en *TTLs*.

Los paquetes generados y recibidos los manipulamos con el framework *Scapy* de *Python* que nos permite acceder a los campos de cada uno y, por

ejemplo, determinar su tipo, *source*, además de poder determinar si hubo o no respuestas por parte del host intermedio.

## 1.2. Métodos

### 1.2.1. Primer ejercicio

Lo primero que necesitamos implementar es el *traceroute* mencionado. Como dijimos antes, se trata de enviar a lo sumo  $30^2$  tandas (hasta que en alguna se reciba un '*echo-reply*') de  $n$  iteraciones (usualmente con  $n = 30$ ), donde para cada tanda se envían paquetes al destino seleccionado con un *TTL* fijo y se va incrementando entre tandas.

Entre envío y respuesta de los paquetes se mide el *RTT* a cada salto para cada iteración con la directiva *time()* de la librería homónima de *Python*<sup>3</sup> (multiplicando por 1000 para medir de a milisegundos).

Para la ruta 'general' tomamos, por cada salto la IP con más apariciones de la tanda y el *RTT* promediado de la misma.

### 1.2.2. Segundo ejercicio

El segundo ejercicio consiste en implementar, sobre la base de la herramienta anterior, una nueva que permita encontrar saltos intercontinentales en la ruta trazada.

Esto se implementa buscando outliers con el método de Cimbala<sup>4</sup> sobre los *RTTs*. La idea es, sobre la distribución  $Z$  del muestreo<sup>5</sup> comparar las mediciones normalizadas  $ZRTT_i$  contra valores de la tabla  $\tau$  de Thompson para un  $\alpha$  dado. Si el valor  $ZRTT_i$  es menor que el valor de la tabla  $\tau$ , entonces se lo asume un *outlier*.

Intuitivamente, los saltos intercontinentales son candidatos a *outlier* dado que se trata de canales con mucho delay.

---

<sup>1</sup><https://en.wikipedia.org/wiki/Traceroute>

<sup>2</sup>Por default, se usa como máxima cantidad de saltos para una *traceroute* 30, si bien el diámetro de Internet es mucho mayor.

<sup>3</sup><https://docs.python.org/3/library/time.html#time.time>

<sup>4</sup><http://www.mne.psu.edu/cimbala/me345/Lectures/Outliers.pdf>

<sup>5</sup>[https://en.wikipedia.org/wiki/Standard\\_score](https://en.wikipedia.org/wiki/Standard_score)

## 2. Mediciones realizadas

Ahora mostraremos los datos obtenidos en todas las mediciones, considerando las herramientas obtenidas en los dos ejercicios de la primera etapa mas un conjunto de gráficos que aporten nuevos datos para poder analizar.

### **3. Conclusiones Generales**