DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 2

Rutas en Internet

7 de diciembre de 2015 — Teoría de las comunicaciones

Integrante	LU	Correo electrónico
González, Sergio Martín	723/10	sergiogonza90@gmail.com
Ladelfa, Hernán Nahuel	318/04	nahueladelfa@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Índice

1. Introducción

El objetivo del siguiente trabajo, es el de estudiar y poder monitorear rutas a nivel de red, y por sobre todo, encontrar rutas hacia hosts en otros continentes, por las que solo se puede acceder a través de enlaces submarinos. La idea es intentar encontrar estos enlaces, e intentar monitorear como se comportan a lo largo del dia.

Para esto, es necesario implementar una herramienta *traceroute* basada en el protocolo ICMP (Internet Control Message Protocol), y que esta se pueda utilizar para la recopilación de los datos necesarios para luego poder hacer los análisis pertinentes. Este protocolo, el cual es el utilizado en la herramienta *ping* de cualquier sistema operativo, es un protocolo de control y notificación de errores, que corre sobre IP. ICMP cuenta con varios tipos de paquetes, pero nosotros solo vamos a enfocarnos en 3:

- ***echo-request***: Paquete utilizado en la herramienta *ping*. Este sirve para saber si un host se encuentra disponible o no. De estar disponible, el host receptor responde con un *echo-reply*.
- ***echo-reply***: Respuesta al envio del tipo de paquete anterior. Cuando un host recibe un paquete de tipo *echo-request*, este envia (de tenerlo habilitado) un paquete de este tipo.
- ***time-exceeded***: Los paquetes ICMP poseen un campo llamado TTL (time to live), el cual indica el tiempo de vida del paquete. Entonces, por ejemplo, si se envia un *echo-request* con un TTL de 3, luego de los 3 saltos de router, este paquete es descartado, y se envía un paquete de tipo *time-exceeded* al nodo que envió el paquete original.

Utilizando esta ultima propiedad de los paquetes ICMP, podemos implementar una herramienta que, envíe paquetes *echo-request* incrementando de a poco el TTL (inicialmente con 1), y quedarnos con las IPs origen de los paquetes *time-exceeded*, asi poder averiguar la IP de cada salto que realiza el paquete al momento de ser enviado. Esto se realiza hasta que se obtiene un paquete de tipo *echo-reply*.

También se deben poder obtener los ΔRTT entre cada hop de la ruta, para poder realizar un test de Grubbs, y encontrar posibles outliers, los cuales son potenciales enlaces submarinos. Para realizar esto, lo que se hace es acumular los RTT de cada respuesta de tipo *time-exceeded*, y luego:

$$\Delta RTT_i = RTT_i - RTT_{i-1}$$

En la etapa de análisis, una ves ya teniendo las la información necesaria, se hara un análisis sobre la posible ubicación de los nodos que se encuentran entrelazados por un enlace submarino. Para esto se utilizara la herramienta Geoiptool [1] con la que se puede geolocalizar una IP.

Luego se realizar un análisis sobre la variación del RTT del enlace submarino, a lo largo del día. Para esto, se realizara un script que corra el traceroute cada 30 minutos, y asi poder realizar un monitoreo sobre la ruta.

[1] www.geoiptool.com/es/

2. Primera Consigna: Caracterizando rutas

2.1. Implementacion de Traceroute

Para el desarrollo de los próximos análsis, se implemento una herramienta *traceroute* sobre Scapy en Python 3. La idea consiste en lo siguiente: Enviar iterativamente paquetes ICMP *echo-request*, empezando con un TTL de 1 y incrementándolo hasta que nos llegue un paquete de tipo *echo-reply*. Por cada paquete enviado, nos guardamos la IP del host originario del paquete, y de esta manera podemos construir la ruta que lleva al mismo.

Sin embargo, hay cosas a tener en cuenta para poder realizar esta técnica. Primero, es posible que por el trafico de la red en el momento de hacer la captura, o por otros motivos que están fuera de nuestro control, el paquete pueda cambiar de ruta en los distintos instantes del envio de los paquetes ICMP. Otra cosa a tener en cuenta, es que si nosotros queremos hacer un estudio sobre el RTT de cada enlace (ΔRTT), no nos sirve el RTT medido para 1 solo intento, ya que esto pudo también haber sido influenciado por el trafico momentáneo de la red.

Para intentar aliviaran esto, lo que hacemos es, no enviar solo 1 paquete por cada TTL, sino que enviamos varios. Cuantos?, los necesarios para poder realizar un promedio lo mas confiable posible del ΔRTT. Lo que hacemos es, establecer una cota inferior y una cota superior a la cantidad de paquetes a enviar de un mismo TTL:

- **MAX_ATTEMPTS**: Cantidad máxima de paquetes **exitosos** a enviar de un mismo TTL. Esto quiere decir que, si mandamos MAX_ATTEMPTS paquetes, y siempre nos llega una respuesta del mismo nodo, entonces promediamos el RRT, anotamos esta IP, y pasamos al próximo TTL.
- **MIN_ATTEMPTS**: Cantidad mínima de paquetes provenientes del mismo nodo que debemos recibir para pasar al próximo TTL. Esto quiere decir, que si nosotros enviamos MIN_ATTEMPTS paquetes, y todos vinieron del mismo nodo, y en MIN_ATTEMPTS + 1, nos viene de un nodo diferente a los anteriores, de todas maneras promediamos los RTT, y avanzamos de TTL. En caso de que la IP cambie antes de llegar a los MIN_ATTEMPTS, se descarta la información recopilada para este TTL y se vuelve a comenzar.

Cabe aclarar que este método igual tiene una falla, y es que si al momento de cambiar la IP, nosotros igual proseguimos al próximo TTL, o mismo cuando no llegamos a los MIN_ATTEMPTS, y debemos reiniciar el TTL, va a haber un ΔRTT que quizá no sea el de 1 enlace físico, ya que la ruta cambio. Por eso es que junto con la información recopilada, nos quedamos también con los intentos que se realizaron para cada salto, así podemos identificar cuando pase esto. Como el objetivo del TP en si es estudiar enlaces submarinos, no es tan importante que la ruta cambie en un momento determinado, siempre y cuando esto no suceda en los extremos del enlace submarino.

2.2. Identificando enlaces Submarinos

Para poder identificar enlaces submarinos, nos basamos en un teste denominado "Test de Grubbs", el cual sirve para detectar outliers en una muestra aleatoria. En nuestro caso, la muestra aleatoria seran los ΔRTT obtenidos en todo el traceroute, y el outlier que queremos encontrar, sera justamente el del enlace submarino, ya que estimamos que este enlace debe tener un RTT mucho mayor a cualquier enlace que se encuentre en tierra.

Para realizar el test de Grubbs, es necesario que la muestra aleatoria sea de distribucion normal, pero de todas maneras, la cátedra indico que igual usemos el test si la distribucion de los ΔRTT no nos queda normal. Ahora para verificar si la distribución de nuestras muestras es normal, utilizaremos el modulo Scipy de Python, el cual permite hacer análisis estadísticos. El test de Grubbs consiste en lo siguiente, dada una muestra X de tamaño n:

1. Se calcula el promedio muestral de X: μ
2. Se calcula la desviacion standard de X: σ
3. Se calcula el estadistico del test: $G = (max(X) - \mu)/\sigma$
4. Se obtiene el valor critico del test: C (Este valor esta tabulado para cada n)
5. Si el valor de G es mayor a C, entonces *max(X)* es un outlier de X

3. Segunda Consigna: Gráficos y Análisis

4. Conclusiones