

Informe

Victor Navajas, Micaela Oliva y Exequiel Muñoz

Informe de Tecnología Digital 4

Información General

- **Curso:** Tecnología Digital 4
- **Fecha:** [29 de octubre de 2023]
- **Profesores:** [Emmanuel Iarussi y Lucio Emilio Santi]
- **Estudiantes:** [Victor Navajas, Micaela Oliva y Exequiel Munoz]

Resumen

En este informe, se abordan tres aspectos del análisis de redes. En la primera parte, implementamos Traceroute utilizando Python y Scapy para rastrear la ruta y los tiempos de respuesta entre un origen y un destino. En la segunda parte, desarrollamos un Port Scanner que detecta el estado de los puertos en un host. Este escáner puede considerar puertos abiertos, cerrados o filtrados y se extiende para verificar la conexión TCP completa. Finalmente, en la tercera parte, realizamos experimentos con estas herramientas en universidades de diferentes continentes y se analizan los resultados, incluyendo un análisis comparativo con la herramienta nmap, adicionalmente vamos a presentar experimentos realizados por nosotros. El informe busca proporcionar una visión integral de estas herramientas y sus aplicaciones en el análisis de redes.

Índice

1. Introducción
2. Explicación de la Implementación
 - Traceroute
 - Port Scanner I
 - Port Scanner II
3. Experimentación
 - 3.1
 - a
 - b
 - 3.2
 - a
 - b
 - c
 - d
4. Conclusiones

1. Introducción

Traceroute... Portscanner...

2. Explicación de la Implementación

2.1 Traceroute

Creamos una función llamada Traceroute donde pasamos como parámetros al host destino y la cantidad máxima de TTL (Time to Live) que tendrá ese paquete. Usando la librería Scapy para cada uno de los TTLs creamos un paquete IP, con su TTL correspondiente hasta llegar a la cantidad máxima. En nuestro caso decidimos definir un TTL igual a 45, que es un poco mayor a lo que está definido en el Traceroute de Windows (TTL=30), esto se debe a que en algunos casos requerimos un número de TTL mayor a 30 porque los hosts de destino estaban muy lejos. Luego enviamos los paquetes. Esperamos las respuestas por 1 segundo para evitar demoras ante una pérdida ya que al hacer varias pruebas observamos que las respuestas no demoraban más de 500 ms. Suponemos que si se supera el tiempo, no hubo respuesta. Adicionalmente tomamos el tiempo con la librería Time para los experimentos. En el caso de que el TTL expira en tránsito (ICMP de tipo 11 y código 0), imprimimos por pantalla la IP del router con el tiempo correspondiente. Por otro lado, también contemplamos el caso en el que si llega al destino. Adicionalmente, en el caso donde no recibamos respuesta, aparece un "**".

2.2 Port Scanner I

Para implementar el primer Port Scanner, nos basamos en un SYN scan, que envía paquetes SYN a los primeros 1000 puertos de un host destino y chequea si responde con un SYN-ACK. Para eso usamos la librería Scapy para crear un paquete IP con un flag SYN de TCP, luego los enviamos. Definimos un tiempo de espera de las respuestas de 500 ms, llegamos a este tiempo luego de hacer varias pruebas y observamos que recibamos un número de respuestas satisfactorio para una cantidad de tiempo razonable (no muy larga y sin mucha diferencia contra tiempos más altos). Para cada paquete enviado, chequeamos si la respuesta tiene los flags SYN y ACK prendidos, en el caso de que se cumpla, consideramos al puerto como “open” o abierto. En el caso contrario, es decir, que no recibamos respuesta o que esté cerrado (la respuesta es negativa) los consideramos “filtered” o filtrado.

2.3 Port Scanner II

A diferencia del primer Port Scanner, para el segundo Port Scanner lo basamos en un CONNECT scan, este funciona de manera similar al primero pero al recibir una respuesta positiva se envía adicionalmente otro paquete ACK con un payload en espera de un ACK. Por lo cual creamos un paquete IP con un flag SYN, lo enviamos en espera de una respuesta con un tiempo de espera de 500 ms por la misma razón que elegimos este tiempo en el primer Port Scanner. En el caso de recibir una respuesta positiva, es decir un paquete SYN-ACK, procedemos a crear un paquete con un ACK y un payload, en el caso contrario consideramos a ese puerto como “filtered”. En nuestro caso decidimos que el payload sea un string de “Hello, [IP del destino!]”. Luego esperamos la respuesta con una espera de 500ms, por lo cual si está llega correctamente, consideramos al puerto escaneado como “open”, en el caso contrario lo consideramos como “filtered”. También implementamos una función llamada Scan Ports que guarda los resultados de ambos Port Scanners en archivos de texto.

3. Experimentación

3.1

a Para el primer experimento elegimos 6 universidades de distintos continentes y son:

- Universidad de Massachusetts-Amherst (América del Norte) - <http://gaia.cs.umass.edu/> (GAIA)
- Universidad China de Hong Kong (Asia) - <http://www.cuhk.edu.hk/> (CUHK)
- Universidad Africana de Ciencia y Tecnología (África) - <http://aust.edu.ng/> (AUST)
- Universidad de Sorbonne (Europa) - <http://www.sorbonne-universite.fr/> (SORBONNE)
- Universidad de San Pablo (América del Sur) - <http://www5.usp.br/> (USP)
- Universidad Torcuato di Tella (América del Sur) - <http://www.utdt.edu/> (UTDT)

Para cada una de las universidades elegidas ejecutamos el traceroute con sus respectivas URLs, implementando el código de `traceroute` tal que nos muestra el camino hecho del paquete por cada router hasta llegar a

su destino y que nos devuelva el porcentaje de paquetes con **ttl-zero-during-transit**, el porcentaje de paquetes de **no-response** y si el paquete termina llegando a su destino o no con **destination-reached**.

Implementamos a **ttl-zero-during-transit** que a partir de un contador que cuenta la cantidad de saltos de las cuales recibimos respuesta exitosa de un router intermedio, a ese valor del contador lo dividimos por la cantidad de hops totales que terminó haciendo ese paquete, en el caso de que haya llegado a su destino podría llegar a ser menor que los 45 hops que habíamos puesto como el máximo, en el otro caso, si no llegaba al destino contaría los 45 hops. Por lo cuál, para calcular el porcentaje de **ttl-zero-during-transit**:

$$\% \text{ ttl - zero - during - transit} = \frac{\# \text{ casos exitosos de los routers intermedios}}{\# \text{ hops totales del paquete}}$$

Por el otro lado, para calcular el porcentaje de hosts intermedios que devolvieron una respuesta negativa para paquetes de **no-response** o no respuesta, es decir, de la cuál no obtuvimos una respuesta del router, sea porque se haya perdido algún paquete o que el router haya decidido de no responder, también inicializamos otro contador que cuente esos casos y lo dividimos por la cantidad total de hops que terminó haciendo el paquete:

$$\% \text{ de no respuesta} = \frac{\# \text{ casos no exitosos}}{\# \text{ hops totales del paquete}}$$

Adicionalmente, si **destination-reached** vale 0 el paquete no terminó llegando al host destino, en cambio si vale 1 es que sí llegó al destino.

Por lo cual, nos pareció interesante arrancar el experimento corriendo la implementación de **traceroute** desde la misma computadora en dos lugares distintos para ver los resultados que podría arrojar, y en qué se podrían diferenciar. Es por eso, que decidimos correrlo desde la universidad con el WiFi de ALUMNOSUTDT, que nos proporcionó estos resultados para cada una de las cinco universidades elegidas:

Host destino	# de hops que hizo	% de ttl-zero-during-transit	% de no recibir respuesta	Llega al destino
GAIA	43	90,70	6,98	1
UTDT	45	0,00	100,00	0
AUST	20	70,00	25,00	1
SORBONNE	13	69,23	23,08	1
CUHK	17	64,71	29,41	1
USP	19	68,42	26,32	1

Figura 1: Resultados de traceroute corrido desde la universidad

Pudimos observar que corriendo desde la universidad, la tasa de éxito era bastante alta, pues de las 6 universidades, los paquetes de 5 de ellas pudieron llegar al host destino, que es un 83,3% de éxito. El único paquete que no llegó a su destino fue el de www.utdt.edu, ya que en todos los 45 hops obtuvimos de respuesta '*', es decir, que tuvo un 100% de no respuesta. Eso podría significar que los routers sobre el path no responden al paquete ICMP o también supusimos que la universidad tiene configurada la firewall para no mostrar esa información en particular por medidas de seguridad, ya que es una página web muy frecuentado por toda la comunidad de UTDT.

Asimismo, observamos que gaia.cs.umass.edu por mas que obtuvo la mayor cantidad de hops que necesito para que llegue al destino, es el host que tuvo el mayor porcentaje de ttl-zero-during-transit con un 90%, mientras que el resto llegaron también a su destino pero cerca de un 70%.

Por el otro lado, lo corrimos también desde la casa de Micaela para ver que resultados nos podría arrojar:

Host destino	# de hops que hizo	% de ttl-zero-during-transit	% de no recibir respuesta	Llega al destino
GAIA	45	6,67	93,33	0
UTDT	45	6,67	93,33	0
AUST	45	4,44	95,56	0
SORBONNE	45	6,67	93,33	0
CUHK	45	6,67	93,33	0
USP	45	6,67	93,33	0

Figura 2: Resultados de traceroute corrido desde la casa de Micaela

En la *Figura 2* podemos ver que los resultados son bastante distintos a los de la *Figura 1*. Una de las razones, es que ninguno de los paquetes llegó a su host destino, tal que hubo 0% de éxito. Aunque, mirando un poco más el path que hace cada uno de los paquetes, vimos que los paquetes en un cierto punto muestran los routers que atravesaron, pero después solamente aparecen los "*". Lo que nos lleva a especular, que los host intermedios y de destino no se quieren comunicar con el host de Micaela por correrlo desde un WiFi personal, en comparación con uno institucional, ya que por ahí no lo ven como un host de origen confiable.

Asimismo, decidimos comparar ambos casos con el tracert de Windows para ver que resultados arroja. Para esto mostraremos que sucedió con las páginas de GAIA y UTDT:

Universidad	
traceroute.py	tracert
Tracing route to gaia.cs.umass.edu	Tracing route to gaia.cs.umass.edu [128.119.245.12] over a maximum of 30 hops:
1: 10.9.255.254 (Time: 12.67 ms)	1 3 ms 1 ms 1 ms 10.9.255.254
2: 200.89.140.129 (Time: 9.00 ms)	2 4 ms 2 ms 2 ms 129-140-89-200.fibertel.com.ar [200.89.140.129]
3: 10.104.63.169 (Time: 6.53 ms)	3 1 ms 6 ms 1 ms 10.104.63.169
4: *	4 * * * Request timed out.
5: 10.100.1.213 (Time: 13.06 ms)	5 3 ms 2 ms 6 ms 10.100.1.213
6: 200.89.164.230 (Time: 15.00 ms)	6 4 ms 1 ms 2 ms 230-164-89-200.fibertel.com.ar [200.89.164.230]
7: 10.110.144.2 (Time: 10.01 ms)	7 2 ms 2 ms 9 ms 10.110.144.2
8: 10.110.144.1 (Time: 6.99 ms)	8 2 ms 2 ms 2 ms 10.110.144.1
9: 200.89.164.229 (Time: 13.00 ms)	9 3 ms 3 ms 5 ms 229-164-89-200.fibertel.com.ar [200.89.164.229]
10: *	10 * * * Request timed out.
11: *	11 361 ms 3 ms * host234.181-96-113.telecom.net.ar [181.96.113.234]
12: 195.22.220.56 (Time: 11.94 ms)	12 3 ms 3 ms 3 ms ae25.baires1.bai.seabone.net [195.22.220.56]
13: 89.221.41.187 (Time: 147.94 ms)	13 130 ms 131 ms 130 ms et2-1-0.miami19.mia.seabone.net [89.221.41.187]
14: 96.87.9.145 (Time: 180.60 ms)	14 129 ms 132 ms 130 ms be-207-pe01.nota.fl.ibone.comcast.net [96.87.9.145]
15: 96.110.36.81 (Time: 136.91 ms)	15 129 ms 128 ms 129 ms be-3101-cs01.miami.fl.ibone.comcast.net [96.110.36.81]
16: 96.110.44.166 (Time: 136.28 ms)	16 130 ms 130 ms 130 ms be-1112-cr12.miami.fl.ibone.comcast.net [96.110.44.166]
17: 96.110.39.249 (Time: 149.10 ms)	17 135 ms 137 ms 135 ms be-303-cr11.jacksonville.fl.ibone.comcast.net [96.110.39.249]
18: 96.110.47.97 (Time: 159.00 ms)	18 140 ms 140 ms 140 ms be-1311-cs03.jacksonville.fl.ibone.comcast.net [96.110.47.97]
19: 96.110.47.110 (Time: 156.83 ms)	19 137 ms 135 ms 136 ms be-1314-cr14.jacksonville.fl.ibone.comcast.net [96.110.47.110]
20: 96.110.39.193 (Time: 163.37 ms)	20 149 ms 149 ms 148 ms be-303-cr11.56marietta.ga.ibone.comcast.net [96.110.39.193]
21: 96.110.32.13 (Time: 155.07 ms)	21 147 ms 147 ms 151 ms be-1311-cs03.56marietta.ga.ibone.comcast.net [96.110.32.13]
22: 96.110.34.234 (Time: 158.37 ms)	22 148 ms 148 ms 150 ms be-1313-cr13.56marietta.ga.ibone.comcast.net [96.110.34.234]
23: 96.110.39.50 (Time: 157.89 ms)	23 150 ms 150 ms 149 ms be-302-cr11.doraville.ga.ibone.comcast.net [96.110.39.50]
24: 96.110.34.173 (Time: 168.43 ms)	24 149 ms 150 ms 154 ms be-1411-cs04.doraville.ga.ibone.comcast.net [96.110.34.173]
25: 96.110.34.206 (Time: 158.14 ms)	25 150 ms 150 ms 150 ms be-1413-cr13.doraville.ga.ibone.comcast.net [96.110.34.206]
26: 96.110.32.2 (Time: 170.30 ms)	26 149 ms 149 ms 149 ms be-301-cr11.beaumeade.va.ibone.comcast.net [96.110.32.2]
27: 68.86.84.29 (Time: 166.25 ms)	27 148 ms 147 ms 147 ms be-1411-cs04.beaumeade.va.ibone.comcast.net [68.86.84.29]
28: 68.86.85.78 (Time: 165.97 ms)	28 149 ms 149 ms 149 ms be-1413-cr13.beaumeade.va.ibone.comcast.net [68.86.85.78]
29: 96.110.36.118 (Time: 169.05 ms)	29 154 ms 152 ms 150 ms be-302-cr12.newark.nj.ibone.comcast.net [96.110.36.118]
30: 96.110.35.81 (Time: 162.77 ms)	30 150 ms 150 ms 150 ms be-1112-cs01.newark.nj.ibone.comcast.net [96.110.35.81]
31: 96.110.42.194 (Time: 164.57 ms)	
32: 162.151.52.226 (Time: 171.01 ms)	
33: 96.108.47.146 (Time: 165.00 ms)	
34: 50.222.38.42 (Time: 160.28 ms)	
35: 192.80.83.109 (Time: 165.59 ms)	
36: 128.119.0.216 (Time: 166.19 ms)	
37: 128.119.7.74 (Time: 161.36 ms)	
38: 128.119.7.66 (Time: 161.61 ms)	
39: 128.119.0.217 (Time: 160.00 ms)	
40: 128.119.3.33 (Time: 168.29 ms)	
41: 128.119.3.32 (Time: 182.00 ms)	
42: 128.119.240.253 (Time: 161.00 ms)	
43: 128.119.245.12 (Destination Reached) (Time: 164.46 ms)	

Figura 3: Resultados de traceroute y de tracert de GAIA corrido desde la universidad

b Para el traceroute de la universidad de EEUU, el salto mas significativo, es decir el que tardo mas comparado con el anterior, tiene una IP de un servidor Italiano, el anterior tambien, esto puede deberse a que los paquetes de datos pueden seguir rutas inesperadas debido a la forma en que se establecen las conexiones entre proveedores de servicios de Internet (ISP) y redes globales. A veces, el enrutamiento puede llevar paquetes a través de ubicaciones geográficas inesperadas.

Para el traceroute de la Universidad China de Hong Kong, el salto mas significativo, es decir el que tardo mas

comparado con el anterior, tiene una IP de un servidor privado, Las redes privadas a menudo implementan políticas de tráfico y filtrado de paquetes. Esto puede causar retrasos en la transmisión de datos y, en consecuencia, un aumento en el RTT.

Para el traceroute de la Africana de Ciencia y Tecnologia ,el salto mas significativo, es decir el que tardo mas comparado con el anterior, tiene una IP de un servidor de Reino Unido, el anterior es de Argentina, los paquetes deben atravesar conexiones internacionales y pasar por infraestructuras de enrutamiento específicas que a veces implican procedimientos de seguridad y verificación adicionales. Estos procesos pueden agregar tiempo al RTT.

Para el traceroute de la Universidad de Sorbonne, el salto mas significativo, es decir el que tardo mas comparado con el anterior, tiene una IP de un servidor privado, Las redes privadas a menudo implementan políticas de tráfico y filtrado de paquetes. Esto puede causar retrasos en la transmisión de datos y, en consecuencia, un aumento en el RTT.

Para el traceroute a la Universidad de San Pablo, el salto mas largo (numero 16) se observa justo despues de tres hops sin respuesta, el aumento en el RTT en el hop número 16 podría deberse a varios factores, como enrutamiento lento, congestión de red o problemas técnicos en ese punto de la ruta. La falta de respuesta en los hops 12, 13 y 14 podría ser el resultado de dispositivos que no responden a las solicitudes de Traceroute.

Para el traceroute de Universidad Torcuato di Tella no se obtuvieron respuestas de ningún host intermedio a lo largo de la ruta hacia el host destino. Esto podría deberse a configuraciones de red que bloquean o no responden a las solicitudes de traceroute, medidas de seguridad para proteger la infraestructura de red o a la inaccesibilidad de la ruta desde la ubicación donde se realizó el traceroute.

3.2

a [Presenta los resultados y análisis de los experimentos relacionados con el Port Scanner, incluyendo un análisis estadístico sobre el porcentaje de puertos abiertos, cerrados y filtrados en servidores de diferentes universidades.]

b [Busca patrones en los estados de los puertos en diferentes servidores y analiza posibles explicaciones para estos patrones.]

c [Compara los resultados del escaneo de servidores con los dos diferentes scanners desarrollados y resalta las diferencias.]

d [Realiza una comparación entre los resultados obtenidos con el Port Scanner desarrollado y la herramienta nmap para el escaneo de servidores.]

4. Conclusiones

[En esta sección, proporciona una conclusión general del trabajo, resumiendo los hallazgos más importantes y su relevancia en el análisis de redes. También puedes mencionar posibles áreas de mejora o futuras investigaciones en el tema.]

\$\$

\$\$