DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Rutas en Internet

Teoría de las Comunicaciones

Integrante	LU	Correo electrónico
Fernández, Gonzalo	836/10	gpfernandezflorio@gmail.com
Aleman, Damián Eliel	377/10	damianealeman@gmail.com
Pizzagalli, Matías	257/12	matipizza@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

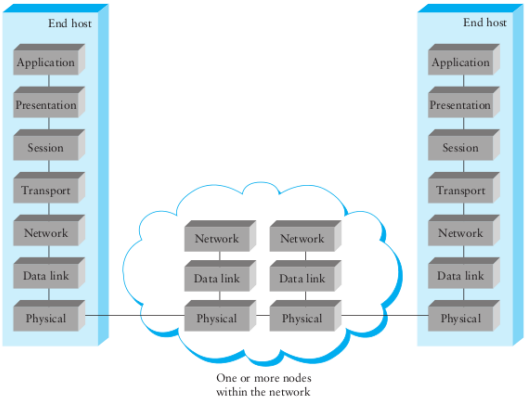
http://www.exactas.uba.ar

Índice

1. Introducción Teórica

1.1. Modelo OSI

El modelo OSI es un modelo que abstrae en 7 capas los protocolos involucrados en la comunicación entre dos nodos en una red.



1.2. Paquetes IP

Los paquetes IP[1] son el método principal de intercambio de mensajes entre nodos a nivel de red. Es decir, cuando los nodos pertenecen a distintas redes locales y no tienen acceso la dirección física del otro. El protocolo IP es *best-effort*, por lo tanto existe la posibilidad de que un mensaje nunca llegue a destino. Es más, los routers que implementan RED[2], están configurados para descartar paquetes periódicamente con alguna probabilidad, incluso antes de entrar en un estado de congestión.

1.3. Paquetes ICMP

Los paquetes ICMP[3] son paquetes de control que no contienen datos, utilizados por los routers para reportar errores en el intercambio de mensajes de una conexión.

1.4. Traceroute

Traceroute es una herramienta que permite mostrar la ruta de una conexión entre dos nodos, identificando todos los nodos intermedios y sus respectivos tiempos de demora. Existen distintas implementaciones de Traceroute a diferentes niveles, que brindan ventajas o desventajas en función de ello. Todas se basan en enviar mensajes con el campo TTL (time to live) incrementándose de uno en uno de forma que, los paquetes sean rechazados por los sucesivos routers en la ruta, al haber finalizado su tiempo de vida. El emisor recoge esas respuestas de los routers y genera el camino virtual correspondiente a partir de las ips

[1]RFC 791 (IP): https://tools.ietf.org/html/rfc791
[2]RFC 2309 (RED): https://tools.ietf.org/html/rfc2309
[3]RFC 792 (ICMP): https://tools.ietf.org/html/rfc792

de los emisores de estos mensajes. Notar que en el caso de los protocolos IP o UDP, donde no se establece una conexión (son orientados a datagramas, no a conexión) es posible que se produzca un cambio de ruta durante el procedimiento. En el momento en que el mensaje llega a destino, se obtiene una respuesta del host objetivo. Esto indica la finalización del Traceroute.

2. Desarrollo

En este trabajo implementaremos una versión de traceroute a través de mensajes ICMP mediante la herramienta Scapy de Python. Como estamos trabajando con paquetes IP, no tenemos asegurado que el paquete enviado llegue a destino, así como que la respuesta llegue de regreso. Debemos tener esto en cuenta al momento de hacer las mediciones.

Mediremos el tiempo de la recepción de una respuesta a un paquete enviado. Sobre estas mediciones utilizaremos la técnica de estimación de outliers propuesta por Cimbala para identificar posibles saltos transcontinentales y un método gráfico para corroborar esa discriminación.

La técnica de estimación de outliers sirve para identificar valores distantes en módulo a la media. Al utilizar esta técnica vamos a estar identificando valores muy por encima y muy por debajo de la media. Sin embargo, como el objetivo es identificar saltos intercontinentales, tendremos en cuenta sólo aquellos outliers por encima de la media, ya que estos presentan una gran distancia entre dos saltos consecutivos.

Al suponer que la carga de la red de una región depende del horario, proponemos como hipótesis que los tiempos totales medidos diferirán según el horario en que se envían debido a distintos niveles de carga en las redes utilizadas. También es posible que esta diferencia no se vea en conexiones lejanas ya que podrían balancearse las demoras entre las distintas regiones. Esto se debe a que tienen distintos usos horarios, luego la carga se distribuye en el tiempo de otra forma.

La implementación se encuentra en el archivo `traceroute.py` y se necesitan permisos de adminstrador para ejecutarlo. Toma como único parámetro opcional la dirección IP (o el nombre) del host destino. A continuación se presenta un pseudocódigo de la misma.

```
1  main(dest = 'www.dc.uba.ar'):
2
3    # Mediciones
4    rtts = []
5    for ttlive in [1..limit]:
6      deltas = []
7      for i in [1..tries]:
8        start = datetime.datetime.now()
9        pkt = sr(IP(dst=dest, ttl=ttlive) / ICMP(), timeout=to, verbose=False)
10       fin = datetime.datetime.now()
11       # RTT hasta el hop actual
12       deltas.append(fin-start)
13
14       # Si cambio de ruta, termino la ejecucion
15       if (cambio_de_ruta())
16         print("Cambio de Ruta!")
```

```
17            sys.exit()
18          # Si ya llegue, dejo de iterar
19          if destino_alcanzado():
20            break
21        rtts.append(avg(deltas))
22
23    relatives = calcular_relativos(rtts)
24    calcular_outliers(relatives)
```

La instrucción principal es el llamado a la función `sr` de Scapy en la línea 9. Esta envía un paquete de control de tipo IP/ICMP con un determinado TTL y devuelve el paquete recibido por respuesta. El valor de TTL itera de uno hasta una constante `limit`. Al recibir cada respuesta, se obtiene un host intermedio (cuya distancia en saltos coincide con el valor de TTL utilizado) en la ruta que se construye. Esto se realiza una determinada cantidad `tries` de veces, para obtener un promedio más confiable entre todos estos intentos.

Como se mencionó anteriormente, el método basado en paquetes IP para implementar Traceroute es propenso a pérdida de paquetes y cambios de ruta. Si no se recibe una respuesta de ninguno de los `tries` paquetes, se considera un salto nulo (probablemente, un router programado para no enviar respuestas de tipo `Time-Exceeded`) y se pasa al siguiente nodo de la ruta (aumentando en uno el TTL). Si se detecta un cambio de ruta (dos emisores distintos en un mismo ciclo TTL) se finaliza la ejecución, ya que no va a ser posible determinar la ruta completa hasta el destino.

Si el emisor de la respuesta es el host al que intentábamos conectarnos, significa que ya tenemos la ruta completa, por lo tanto no hace falta seguir iterando el valor de TTL. Estimamos que no vamos a encontrar una ruta con más hosts intermedios que el valor máximo de TTL, es decir `limit`.

Finalmente se calculan los outliers usando el procedimiento de estimación de outliers mencionado anteriormente. Este toma un arreglo con los valores de RTT relativos de cada nodo no nulo de la ruta. Llamamos RTT relativo de un nodo al valor de RTT de dicho nodo, restándole el valor de RTT del nodo anterior. De esta forma, son los valores de RTT relativos muy altos los que queremos identificar como outliers, ya que estos se corresponderán con nodos detrás de un enlace intercontinental.

Adicionalmente, se imprimen por pantalla cada una de las direcciones IP obtenidas, junto con el RTT estimado desde el origen, la diferencia respecto al RTT del nodo anterior y la ubicación geógrafica del router obtenida a partir de `http://freegeoip.net/`[4]. También se imprimen los outliers estimados, junto con su valor de RTT relativo. Esto es necesario ya que sólo nos interesan los outliers con valor de RTT relativo positivo.

[4]Para más información sobre la implementación de la herramienta automática de obtención de posiciones geográficas, a partir de la dirección IP, consultar el apéndice

3. Resultados

3.1. The University of Tokio

Acontinuación mostramos los resultados obtenidos cuando tomamos como destino la Universidad de Tokio (www.u-tokyo.ac.jp).

HOP	RTT	RTT RELATIVO	Ubicación	ZRTT
192.168.1.1	62 ms	0	-	-0,3018867925
10.21.192.1	159 ms	97	-	1,528301887
10.242.1.17	123 ms	-36	-	-0,9811320755
195.22.220.33	76 ms	-47	(Italy)	-1,188679245
195.22.220.32	127 ms	51	(Italy)	0,6603773585
195.22.219.145	160 ms	33	(Italy)	0,320754717
195.22.219.145	95 ms	-65	(Italy)	-1,528301887
149.3.181.65	106 ms	11	(Italy)	-0,09433962264
129.250.2.227	222 ms	116	Englewood (United States)	1,886792453
129.250.4.13	272 ms	50	Englewood (United States)	0,641509434
129.250.2.54	269 ms	-3	Englewood (United States)	-0,358490566
129.250.3.86	418 ms	**149**	Englewood (United States)	**2,509433962**
129.250.6.188	405 ms	-13	Englewood (United States)	-0,5471698113
129.250.2.255	404 ms	-1	Englewood (United States)	-0,320754717
61.200.80.218	396 ms	-8	(Japan)	-0,4528301887
158.205.192.173	414 ms	18	(Japan)	0,03773584906
158.205.192.86	413 ms	-1	(Japan)	-0,320754717
158.205.121.250	387 ms	-26	(Japan)	-0,7924528302
154.34.240.254	407 ms	20	(Japan)	0,07547169811
210.152.135.178	399 ms	-8	(Japan)	-0,4528301887

4. Conclusiones

5. Apéndice