# Redes de computadoras 2015
# Obligatorio 1

**Grupo 48**
Alonzo Fulchi, Emiliano
Damiano Izzi, Rodrigo Sebastian
Flores Saavedra, Martin

Tutor: Eduardo Grampín

# 1 - Comando Ping

1) Ping es una herramienta de diagnóstico de redes de computadoras que se utiliza para comprobar la accesibilidad de un host en una red IP y para medir el RTT de los mensajes enviados. Ping funciona enviando paquetes ICMP echo request a un host y espera por un paquete ICMP echo reply, midiendo el tiempo desde que se envía el paquete y se recibe una respuesta. Luego de terminar su ejecución, se obtiene un resumen estadístico del resultado de la ejecución como por ejemplo el porcentaje de paquetes perdidos, el RTT mínimo, promedio y máximo, entre otros.

2) Para esta parte tomamos como métrica el RTT promedio. Nos pareció razonable ya que lo que se está midiendo es el host que mejor tiempo de respuesta tiene, por lo que si tomáramos el máximo o el mínimo no necesariamente la respuesta sería la correcta. Un ejemplo al usar el máximo sería que un host devolviera un valor de RTT muy grande solo una vez, y que todos los demás valores fueran menores a todos los tiempos de RTT del otro host. Si se tomara el máximo cómo métrica el resultado sería que el segundo host tiene un mejor tiempo, lo que no consideramos correcto. El mismo problema surge (de manera inversa) al utilizar el mínimo.
   Al script se le debe introducir como parámetros los dos host y opcionalmente la cantidad de paquetes que se van a enviar por cada ejecución de ping.

3) El script para resolver el problema lo realizamos en Python, dado que comenzamos con problemas para manejar un loop en bash. Recibe como parámetros obligatorios dos host y opcionalmente la cantidad de paquetes que se envían por ejecución de ping y el umbral de espera en milisegundos. Una vez definidos los parámetros, el programa entra en un loop infinito que realiza las siguientes acciones: Ejecutar ping y parsear la salida en busca de errores, en caso de encontrar algún error lo imprime en pantalla. Estas acciones se realizan para cada uno de los hosts. Los errores pueden ser:
   a) El destino es desconocido en caso de que ping imprima “unknown host”
   b) El destino es inalcanzable en caso de “Destination Host Unreachable”
   c) No se han recibido paquete de respuesta en caso que se pierdan todos los paquetes
   d) Se ha superado el umbral de espera, en caso que al procesar la salida, el tiempo máximo de un paquete sea superior al umbral

4) Luego de realizar pruebas con varios dominios (i.e www.google.com, www.rediris.es, www.debian.org) no se notaron diferencias significativas entre los tiempos de RTT usando y no usando la bandera -n. Al usar la bandera el comando ping no imprime el nombre canónico del host al que quiere llegar. En una primera instancia se pensó que al usar la bandera el tiempo iba a ser menor ya que el comando no tendría que resolver el nombre canónico del host. Al analizar el tráfico con Wireshark se observó que la diferencia entre ambos comandos es la de realizar una consulta DNS más, para obtener el nombre canónico del host a partir de la IP. Cabe destacar, además, que los tiempos que imprime el comando ping son los de RTT de cada paquete echo_request por lo que

las consultas DNS no deberían afectarlos.

5) Tamaño de las pruebas
    a) Según la flag -s que se ve en el man del comando, el paquete tiene un tamaño por defecto de 56 bytes (que se convierten en 64 bytes cuando se agrega la cabecera del paquete ICMP). El comando no especifica un tamaño máximo por defecto, pero se tiene que considerar que el ping tiene que enviar un solo paquete, lo que implica que no se puede superar el tamaño máximo de datos que entran en un paquete IP. Para especificar este tamaño se utilizan 16 bits, lo que nos da un tamaño máximo de 64KB.

    b) Luego de realizar varias pruebas, no encontramos una variación significativa en los tiempos cuando se envían paquetes de 100 bytes o 1000 bytes. La diferencia grande que encontramos fue entre estos y los paquetes de 10.000 bytes. Se puede apreciar una diferencia casi del doble del tiempo en algunos casos. Esto podría deberse a que cuando los paquetes son demasiado grandes, si bien no superan los 64KB, es muy probable que estos paquetes sean fragmentados, lo que implicaría más tiempo de procesamiento y retardo en las colas, por lo que es razonable que demoren más tiempo.
    A continuación se muestra el resultado de una de las pruebas que hicimos enviando 500 paquetes a www.google.com, se puede apreciar que el RTT promedio de las pruebas con paquetes de 100 y 1.000 bytes es prácticamente el mismo (8.399 ms y 8.992 ms respectivamente) y que el RTT promedio de las pruebas con paquetes de 10.000 bytes es un poco menos que el doble (14.515 ms).

```
paquetes de tamaño 100 bytes
--- www.google.com ping statistics ---
500 packets transmitted, 500 received, 0% packet loss, time 499910 ms
rtt min/avg/max/mdev = 1.991/8.399/210.781/16.042 ms

paquetes de tamaño 1.000 bytes
--- www.google.com ping statistics ---
500 packets transmitted, 499 received, 0% packet loss, time 499928 ms
rtt min/avg/max/mdev = 2.824/8.992/110.787/13.847 ms

paquetes de tamaño 10.000 bytes
--- www.google.com ping statistics ---
500 packets transmitted, 498 received, 0% packet loss, time 499929 ms
rtt min/avg/max/mdev = 7.473/14.515/101.927/11.156 ms
```

# 2 - Comando Traceroute

1) Traceroute, al igual que ping, es una herramienta de diagnóstico de redes de computadoras utilizada para “trackear” la ruta que toman los paquetes de una red IP hacia un host. Utiliza el campo TTL del protocolo IP e intenta producir una respuesta ICMP TIME_EXCEEDED por cada gateway a lo largo de la ruta.
   Traceroute intenta rastrear la ruta que toma un paquete IP, a través de la red hacia un host, enviando paquetes de prueba con un TTL pequeño y espera por una respuesta ICMP TIME_EXCEEDED desde un gateway. Inicialmente las pruebas se envían con TTL 1 y se incrementa en uno, hasta que se obtenga una respuesta ICMP “port unreachable” que significa que se alcanzó el host, o hasta que se alcanza el máximo TTL (su valor por defecto es 30). En cada configuración del TTL se envían 3 pruebas, y se imprime una línea que contiene, el TTL, la dirección del gateway y el RTT de cada prueba. En caso de no recibir una respuesta después de un tiempo (por defecto 5s) se imprime un asterisco (*).

2) Para este programa se decidió definir una variable en el script (llamada umbral) que define cuánto es el tiempo significativo de un hop. El script calcula el tiempo promedio de cada paso y evalúa si la diferencia entre el promedio del paso anterior y el actual supera el umbral definido. En caso de superarlo imprime los nombres de los hosts involucrados.

3) Dado que no hay un forma directa para obtener el país de origen de una IP a través de la información que brinda el traceroute y teniendo en cuenta que un tiempo “grande” no necesariamente implica que se trate de un ISP internacional. Se decidió armar una lista de los nombres de distintos ISPs y alguna expresión que identifique a los hosts del ISP. Esta lista se guardó en un archivo, cada elemento de la lista es de la forma <NOMBRE_ISP>,<EXPRESION_HOST>.

4) Los servidores Looking Glass son servidores que ofrecen herramienta de diagnósticos de redes (i.e traceroute, ping) a través de un portal web. En general están “hosteadas” por ISP en distintos lados del mundo. Estos servidores permiten ejecutar los mismos comandos en distintas partes del mundo, dando la posibilidad de tomar las mismas métricas (por ejemplo hacia un host propio) de distintos lugares e identificar problemas de conectividad causados por ISP en particular o una zona en especial.

   Para las pruebas utilizamos los servidores Princeton University (AS88) (LG1) y Cogent (AS174) (LG2) . Para el segundo nos solicita, ademas del nombre o IP del host de destino, la localización del router a partir del cual realizará la prueba. En nuestro caso seleccionamos US - San Francisco.

   El resultado obtenido en LG1 es el siguiente

```
traceroute to 38.100.128.10 (38.100.128.10), 30 hops max, 40 byte packets
 1  core-87-router (128.112.128.2)  0.563 ms  0.554 ms  0.547 ms
 2  border-87-router (128.112.12.142)  0.427 ms  0.353 ms  0.378 ms
 3  te0-0-1-1.204.rcr12.phl03.atlas.cogentco.com (38.122.150.1)  3.476 ms  3.534 ms
```

```
2.650 ms
 4  te0-0-1-3.rcr21.phl01.atlas.cogentco.com (154.54.27.117)  3.609 ms te0-0-1-
3.rcr22.phl01.atlas.cogentco.com (66.28.4.233)  4.735 ms  4.716 ms
 5  te0-8-0-2.ccr41.dca01.atlas.cogentco.com (154.54.42.89)  7.586 ms te0-8-0-
2.ccr42.dca01.atlas.cogentco.com (154.54.42.101)  6.990 ms  7.849 ms
 6  te4-2.ccr01.iad03.atlas.cogentco.com (154.54.29.122)  201.036 ms te2-
8.ccr01.iad03.atlas.cogentco.com (154.54.41.254)  215.435 ms te4-
2.ccr01.iad03.atlas.cogentco.com (154.54.29.122)  209.742 ms
 7  te2-1.mag01.iad03.atlas.cogentco.com (154.54.86.62)  324.812 ms  217.515 ms
14.419 ms
 8  cogentco.com (38.100.128.10)  8.384 ms !X  9.895 ms !X  8.457 ms !X
```

El resultado obtenido en LG2 es el siguiente

```
traceroute to www.net.princeton.edu (128.112.128.55), 30 hops max, 60 byte packets
 1  vl3.mag02.sfo01.atlas.cogentco.com (66.250.250.145)  0.407 ms  0.408 ms
 2  te0-7-0-14.ccr22.sfo01.atlas.cogentco.com (154.54.47.37)  0.745 ms te0-7-0-
14.ccr21.sfo01.atlas.cogentco.com (154.54.47.33)  0.653 ms
 3  be2133.ccr22.mci01.atlas.cogentco.com (154.54.30.66)  38.652 ms  38.661 ms
 4  be2157.ccr42.ord01.atlas.cogentco.com (154.54.6.118)  50.815 ms  50.789 ms
 5  be2185.ccr22.cle04.atlas.cogentco.com (154.54.43.178)  57.628 ms
be2351.ccr21.cle04.atlas.cogentco.com (154.54.44.86)  58.620 ms
 6  be2482.ccr41.jfk02.atlas.cogentco.com (154.54.27.158)  70.602 ms
be2483.ccr42.jfk02.atlas.cogentco.com (154.54.29.202)  70.595 ms
 7  be2333.rcr22.phl01.atlas.cogentco.com (154.54.5.2)  72.729 ms  73.538 ms
 8  te0-0-2-1.rcr12.phl03.atlas.cogentco.com (154.54.27.118)  73.238 ms te0-0-2-
0.rcr12.phl03.atlas.cogentco.com (66.28.4.234)  73.287 ms
 9  38.122.150.2 (38.122.150.2)  75.379 ms  75.686 ms
10  core-87-router.Princeton.EDU (128.112.12.130)  75.022 ms  75.457 ms
11  www.net.Princeton.EDU (128.112.128.55)  76.281 ms  75.532 ms
```

La primera diferencia que notamos es que el tamaño de los paquetes que utilizan es diferente. Cuando se ejecuta el comando desde el LG1 el tamaño de los paquetes es de 40 bytes y cuando se ejecuta el comando desde el LG2 el tamaño de los paquetes es de 60 bytes.
Otra diferencia es que desde LG1 se envían 3 paquetes de prueba y desde LG2 solo se envían 2.
Otra diferencia que hay que notar, si miramos el origen y destino de cada prueba, los IPs de los hops intermedios deberían ser parecidos (están en la misma red). Se puede ver que el IP destino del resultado de LG2 (128.112.128.55) es parecido al IP del primer hop del resultado de LG1 (128.112.128.2), sin embargo, si miramos el IP destino del resultado de LG1 (38.100.128.10) no se parece al primer hop del resultado de LG2 (66.250.250.145). Esto se debe a que en LG2 nos da la opción de elegir una ciudad donde se encuentra el router del cual parten los paquetes de prueba (US - San Francisco).
La diferencia que encontramos más relevante es que los caminos que toman los paquetes de prueba en un sentido y otro, son totalmente diferentes, lo cual es razonable ya que el ruteo de los paquetes depende del estado de la red en el momento en que se

envían.

# 3 - Comando Dig

1) Teniendo en cuenta que el sistema DNS es el que se encarga de resolver los nombres de los hosts, es decir realiza la traducción a direcciones IP, una posible causa podría ser que el dueño del sitio web al que el usuario quiere acceder haya cambiado la IP del sitio web y el servidor de DNS tenga el registro desactualizado. También a nivel de sistema operativo es posible configurar la IP del servidor del DNS, si esta dirección es errónea cualquier intento de acceder a un sitio web (a través de un nombre) resultará en un error.
2) El script realiza los siguientes pasos:

```
- Ejecuta el comando dig con la flag NS para obtener el nombre de algún
servidor autoritativo (SA).
- Se parsea la línea de flags para obtener la cantidad de respuestas.
- Si la cantidad de respuestas es 0
    1. Se busca en la línea que contiene la etiqueta SOA el dominio del
       host.
    2. Se ejecuta el comando dig nuevamente pero contra el dominio obtenido.
- Se parsea la respuesta obtenida en busca de SA
- Se imprimen SA en pantalla
```

3) El Script realiza:

- Se utiliza el algoritmo anterior y se elige uno de los servidores autoritativos.

- Se realiza ***dig @servidorAutoritativo host RR*** siendo RR el valor pasado por parámetro

- Se imprime el resultado

4) El Siguiente script

- Se utiliza el algoritmo 3-2 y se elige uno de los servidores autoritativos.

- Se realiza ***dig @servidorAutoritativo host MX***

- Luego para cada servidor de la respuesta en el paso anterior se obtiene la ip

ejecutando *dig @servidorAutoritativo hostObtenidoEnPaso2 A*

- Finalmente teniendo la lista de ips de host de servidores de correo se chequea si la **ip** pasada por parámetro pertenece a dicha lista.

5) La diferencia entre la respuesta de los comandos está en la cantidad de registros que son devueltos. Para www.fing.edu.uy se devuelve sólo un registro, mientras que para www.google.com se devuelven doce registros. Dado que www.google.com tiene que atender muchos más solicitudes que www.fing.edu.uy, es coherente que tenga varios servidores que atiendan sobre el mismo nombre. Al tener varias IPs google utiliza cómo mecanismo de balanceo de carga el propio servidor DNS, ya que al ir expirando los registros, el primero que devuelve va a ir rotando, de esta manera para distintos accesos se utilizan distintos hosts.

6) Al realizar la consulta con las direcciones de Google (www.google.com y google.com) ambas pruebas devolvieron varios registros (por lo explicado en la parte 4). Esto es porque Google considera que se sobreentiende que al acceder a google.com se está accediendo a su sitio web (algo que es explícito al usar www.google.com). Por otro lado, al ejecutar la línea “dig fing.edu.uy” observamos que no se retorna ningún registro, cosa que si sucede para “dig www.fing.edu.uy”. Esto implica que no existe un host público con el nombre fing.edu.uy. Generar ambos nombres es habitual entre empresas y/o instituciones (i.e Facebook, ORT, Presidencia, Montevideo COMM). Consideramos que no contar un IP asociada al nombre sin el prefijo www puede generar una “mala imagen”, ya que un usuario al acceder con su navegador a por ejemplo fing.edu.uy se encontrará con un error de que no se pudo resolver el nombre del servidor quizás llevándolo a pensar que el servicio no está disponible. No observamos una desventaja de mantener ambos nombre más que el simple hecho de mantenerlos.

# 4 - Captura de tráfico con Wireshark

1) Wireshark es una herramienta que nos permite realizar capturas de tráfico en redes de computadoras. Nos permite ver lo que está pasando en una red con lujo de detalle. Estas capturas brindan información detallada de los mensajes utilizados por los diferentes protocolos que se utilizan (i.e. ICMP, TCP, UDP, etc.) Las capturas se pueden realizar en tiempo real, lo que nos permite interactuar y analizar el tráfico que hay en cualquier momento. También permite guardar capturas realizadas de forma de poder analizarlas en otro momento.

2) En la sección 3, todos los programas utilizan el comando dig para realizar su trabajo. Teniendo en cuenta que este comando utiliza el protocolo DNS para realizar su trabajo, lo primero que se hizo fue aplicar un filtro para el protocolo DNS. Lo primero que se observa es que realiza la consulta por el registro del tipo A que contiene a "www.fing.edu.uy", luego no se hacen más consultas en cuanto a la información que se devuelve en la consulta. En el paso 32 se busca al registro de tipo A con "gnu.org", inmediatamente a la respuesta (en el paso 42) se consulta por la IP del dominio "ns1.gnu.org" (servidor autoritativo de "gnu.org"). Otra vez en el paso siguiente de la respuesta, se pide por los registros de servidores de correo (registros del tipo MX) y esta vez el pedido se hace al servidor autoritativo. Este último paso nos lleva a concluir que se trata de una captura de la sección 4. En la captura lo que se ve es el procedimiento para obtener los servidores de correo de "gnu.org" desde un servidor autoritativo. Una vez que ya se sabe cuáles son los servidores se los puede comparar al parámetro de entrada. Un detalle a considerar es que en la captura no encontramos en qué momento se busca el servidor autoritativo de gnu.org (ns1.gnu.org) al cual se le solicita posteriormente el registro MX.

3) En la sección 2, todos los programas utilizan el comando traceroute, que como se mencionó antes, este envía varios paquetes de prueba utilizando el protocolo UDP, el cual necesita la dirección IP de destino entre otras cosas para saber a quién enviar los paquetes. Para saber esta dirección IP, dadas las condiciones de la sección 2 hay dos formas, o usamos el IP que se ingresa por parámetro o utilizamos el protocolo DNS para obtener la dirección IP del FQDN. Entonces, para saber si se utilizó un FQDN o una dirección IP lo que hicimos fue aplicar un filtro a la captura especificada, de forma de solo ver los mensajes DNS. Luego buscamos si había algún mensaje DNS donde se solicitara el registro A de algún FQDN. En caso de encontrar este mensaje, se puede ver que luego hay un mensaje DNS respondiendo a esta solicitud donde nos envían la dirección IP de este FQDN. Para confirmar que el destino del comando traceroute es el FQDN del cual averiguamos la dirección IP, nos fijamos si los mensajes UDP, correspondientes a la ejecución del comando traceroute, se envían con destino la IP del FQDN que mencionamos antes. Para esto filtramos la captura por el protocolo UDP y nos fijamos el destino de los mismos. Otro detalle que también verificamos fue que los mensajes UDP enviados a esta dirección IP fueran incrementando el TTL cada tres mensajes y así fue.

   El FQDN que se utilizó fue www.universidad.edu.uy y el IP que se obtuvo a través del protocolo DNS fue 164.73.2.136.

# 5 - Referencias


[1] Wireshark - https://www.wireshark.org/
[2] Dig - http://linux.die.net/man/1/dig

[3] Traceroute - http://linux.die.net/man/8/traceroute
[4] Ping - http://linux.die.net/man/8/ping
[5] Lista de ISPs - http://as-rank.caida.org/
[6] Python - http://www.tutorialspoint.com/python/index.htm
[7] Bash - http://www.tutorialspoint.com//unix_commands/bash.htm