Redes de Computadoras

Curso 2017

Grupo 76

Informe - Obligatorio 01

Autores:
Joaquín Villar
Federico Luongo
Federico Godoy

Supervisores:
Martín Giachino
Jorge Visca

August 27, 2017

Contents

1 Parte A - Comando ping

1.1 Funcionamiento del comando

El comando Ping permite verificar el nivel de conexión entre un host origen y un host destino identificado por determinada IP. Para ello se envian paquetes ICMP de solicitud (Internet Control Message Protocol Echo Request) y de respuesta (ICMP Echo Reply), en está se puede obtener información sobre la conectividad como por ejemplo el estado , velocidad y calidad de una red determinada.
Ejecutando Ping de solicitud, el Host local envía un mensaje ICMP, contenido en un paquete IP. El mensaje ICMP de solicitud incluye, además del tipo de mensaje y el código del mismo, un número identificador y una secuencia de números, de 32 bits, que deberán coincidir con el mensaje ICMP de respuesta; además de un espacio opcional para datos. Como el protocolo ICMP no se basa en un protocolo de capa de transporte como TCP o UDP no utiliza ningún protocolo de capa de aplicación.
El Ping trabaja en la capa de red del protocolo TCP/IP y es un tipo de mensaje de control del protocolo ICMP, sub protocolo IP.

1.1.1 Parte a

EL comando utiliza los protocolos ICMP Echo Request y ICMP Echo Reply. El Protocolo de Mensajes de Control de Internet o ICMP (por sus siglas en inglés de Internet Control Message Protocol) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un router o host no puede ser localizado. También puede ser utilizado para transmitir mensajes ICMP Query.

El Echo Request es un mensaje de control que se envía a un host con la expectativa de recibir de él un Echo Reply. Esto es conocido como Ping y es una utilidad del protocolo ICMP, subprotocolo de IP. Todo host debe responder a un Echo Request con un Echo Reply que contenga exactamente los mismos datos que el primero. Un Echo Reply (Respuesta de Eco) en el protocolo ICMP es un mensaje generado como respuesta a un mensaje Echo Request (petición de Eco).

1.1.2 Parte b

En la salida del comando Ping se obtienen los siguientes datos:

- La dirección IP del host destino
- El número de secuencia ICMP
- El valor de TTL
- El valor de RTT
- El tamaño en bytes del paquete enviado

El campo de vida útil (TTL) permite conocer la cantidad de routers por los que pasó el paquete mientras viajó de una máquina a otra. Cada paquete IP posee un campo TTL inicialmente alto, como máximo 255. Cada vez que pasa por un router, se reduce el valor en uno. Este valor también se decrementa luego de cierto tiempo si el paquete no avanza en su camino. Si alguna vez este número es cero, el router interpretará que el paquete está viajando en círculos, por lo tanto, finaliza el proceso.

El campo de demora de vueltas (RTT) corresponde al lapso de tiempo en milisegundos que se necesita para dar una vuelta entre las máquinas fuente y destino. Cómo regla general, la demora de un paquete no debe ser mayor a 200 ms.

1.2 Ejecución del comando

```
[joaquin.villar@pcunix140 ~]$ ping -c 5 www.antel.com.uy
PING portalweb.antel.com.uy (190.0.136.235) 56(84) bytes of data.
64 bytes from r190-0-136-235.ir-static.adinet.com.uy (190.0.136.235): icmp_seq=1 ttl=52
64 bytes from r190-0-136-235.ir-static.adinet.com.uy (190.0.136.235): icmp_seq=2 ttl=52
64 bytes from r190-0-136-235.ir-static.adinet.com.uy (190.0.136.235): icmp_seq=3 ttl=52
64 bytes from r190-0-136-235.ir-static.adinet.com.uy (190.0.136.235): icmp_seq=4 ttl=52
64 bytes from r190-0-136-235.ir-static.adinet.com.uy (190.0.136.235): icmp_seq=5 ttl=52

--- portalweb.antel.com.uy ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.485/2.036/3.999/0.983 ms
[joaquin.villar@pcunix140 ~]$ ping -c 5 www.google.com
PING www.google.com (172.217.28.196) 56(84) bytes of data.
64 bytes from eze03s30-in-f4.1e100.net (172.217.28.196): icmp_seq=1 ttl=54 time=18.5 ms
64 bytes from eze03s30-in-f4.1e100.net (172.217.28.196): icmp_seq=2 ttl=54 time=16.0 ms
64 bytes from eze03s30-in-f4.1e100.net (172.217.28.196): icmp_seq=3 ttl=54 time=16.0 ms
64 bytes from eze03s30-in-f4.1e100.net (172.217.28.196): icmp_seq=4 ttl=54 time=16.1 ms
64 bytes from eze03s30-in-f4.1e100.net (172.217.28.196): icmp_seq=5 ttl=54 time=16.0 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 16.074/16.595/18.581/0.996 ms
[joaquin.villar@pcunix140 ~]$ ping -c 5 registro.br
PING registro.br (200.160.2.3) 56(84) bytes of data.
64 bytes from registro.br (200.160.2.3): icmp_seq=1 ttl=240 time=44.1 ms
64 bytes from registro.br (200.160.2.3): icmp_seq=2 ttl=240 time=41.9 ms
64 bytes from registro.br (200.160.2.3): icmp_seq=3 ttl=240 time=42.0 ms
64 bytes from registro.br (200.160.2.3): icmp_seq=4 ttl=240 time=41.9 ms
64 bytes from registro.br (200.160.2.3): icmp_seq=5 ttl=240 time=41.9 ms

--- registro.br ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 41.944/42.406/44.164/0.907 ms
[joaquin.villar@pcunix140 ~]$ ping -c 5 zadna.org.za
PING zadna.org.za (129.232.249.120) 56(84) bytes of data.
64 bytes from www520.jnb1.host-h.net (129.232.249.120): icmp_seq=1 ttl=46 time=416 ms
64 bytes from www520.jnb1.host-h.net (129.232.249.120): icmp_seq=2 ttl=46 time=416 ms
64 bytes from www520.jnb1.host-h.net (129.232.249.120): icmp_seq=3 ttl=46 time=415 ms
64 bytes from www520.jnb1.host-h.net (129.232.249.120): icmp_seq=4 ttl=46 time=415 ms
64 bytes from www520.jnb1.host-h.net (129.232.249.120): icmp_seq=5 ttl=46 time=415 ms

--- zadna.org.za ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
```

Figure 1: Impresión de la prueba.

1.2.1 Parte a

Dado que el el tiempo de respuesta se obtiene en el campo de demora de vueltas (RTT) que corresponde al lapso de tiempo en milisegundos en ir y venir de la máquina fuente a destino. Se observa que el servicio que tiene menor demora es el que corresponde al host 'Antel', este tiempo se encuentra entre 1.5 y 3.9 ms. Esto era de esperar dado que la distancia a la ubicación geográfica al host es la menor de todas y entonces los paquetes deben realizar un menor recorrido mientras que el resto de los hosts ubicados a distancias mayores aumentan el valor de RTT dado que los paquetes deben realizar un mayor recorrido a través de la red. En

segundo lugar se tiene al servicio correspondiente a 'Google' seguido por el servicio 'Registro.br'.
Por último y con tiempos de respuesta considerablemente mayores se tiene al servicio 'zadna', en el orden de los 400 ms. Dada la gran diferencia obtenida en el último host, mediante la herramienta web de geolocalizacion según la direccion ip (www.iplocation.net) se puede verificar que el host está ubicado en Sudáfrica, el cual se corresponde con los valores obtenidos de RTT para dicho host.

1.2.2 Parte b

Dadas las salidas obtenidas, se obtienen los valores de TTL siguientes en cada caso:

Registro.br: TTL = 240
Google: TTL = 54
Antel: TTL = 52
Zadna: TTL = 46

EL valor de TTL final depende del valor inicial asignado por el sistema operativo del host destino, por esto el valor TTL que nosotros recibimos es aquel que fue seteado por el host destino menos la cantidad de saltos hasta llegar a nosotros. Entonces no es posible hacer comparaciones en cuanto a cantidad de saltos entre los hosts dado que para cada host destino el valor de TTL asignado a cada paquete puede ser distinto. Analizando los valores se puede observar que el TTL de la dirección Registro.br inicial asignado por ese host es 255 y el valor de TTL que se obtiene es de 240 entonces la cantidad de saltos a dicho host es de 15. Con respecto a los demás hosts, los valores de TTL son todos inferiores al primero pero no se puede concluir con seguridad que los TTL iniciales asignados por dichos hosts sean iguales o similares. Suponiendo cierto lo anterior se puede analizar que los resultados son similares en cantidad de saltos entre los hosts Google y Antel. Mientras que la cantidad de saltos del Host Zadna ubicado en Sudáfrica es mayor a los anteriores. Como alternativa a lo explicado anteriormete para poder contestar cual es el host mas lejano en candidad de saltos se decidió ejecutar el comando ping nuevamente para cada host haciendo uso de la flag -i que permite setear el valor de TTL inicial. De esta forma ejecutando varias veces el comando y comenzando

con TTL en 1 observamos que obtenemos como respuesta Time to Live Exceded, a medida que incrementamos el valor en sucesivas ejecuciones hasta obtenener una respuesta válida pudimos concluir cual es la distancia en hops a cada host. Luego de este experimento se obtienen los siguientes resultados:

Antel: 7 hops
Zadna: 14 hops
Registro.br: 14 hops
Google: 7 hops

1.2.3 Parte c

Para cada instancia de la ejecución del comando se observan cambios en los valores de RTT, aunque estos son mínimos se puede observar como patrón que la primer instancia siempre tiene un valor mas elevado en tiempo de respuesta respecto al resto de las instancias. Está diferencia de tiempo entre las primeras instancias y las restantes se puede justificar en base a tener un tiempo extra cuando se realiza un DNS lookup mientras que en las posteriores instancias este tiempo no se tiene dado que luego de la primer request esto queda cacheado.

1.2.4 Parte d

Al ejecutar el mismo comando sucesivas veces, no se obtienen grandes diferencias dado que el lapso de tiempo entre la ejecución de un comando y otro es relativamente corto. En un período corto la situación de la red no sufre grandes cambios, para poder observar diferencias ejecutamos el comando más de una vez con una diferencia de horas mayor. Luego de esto observamos que mientras en cierto horario el tiempo de respuesta (RTT) es más lento que en otros debido a cierta congestión en la red los valores de TTL se mantienen como era de esperar.

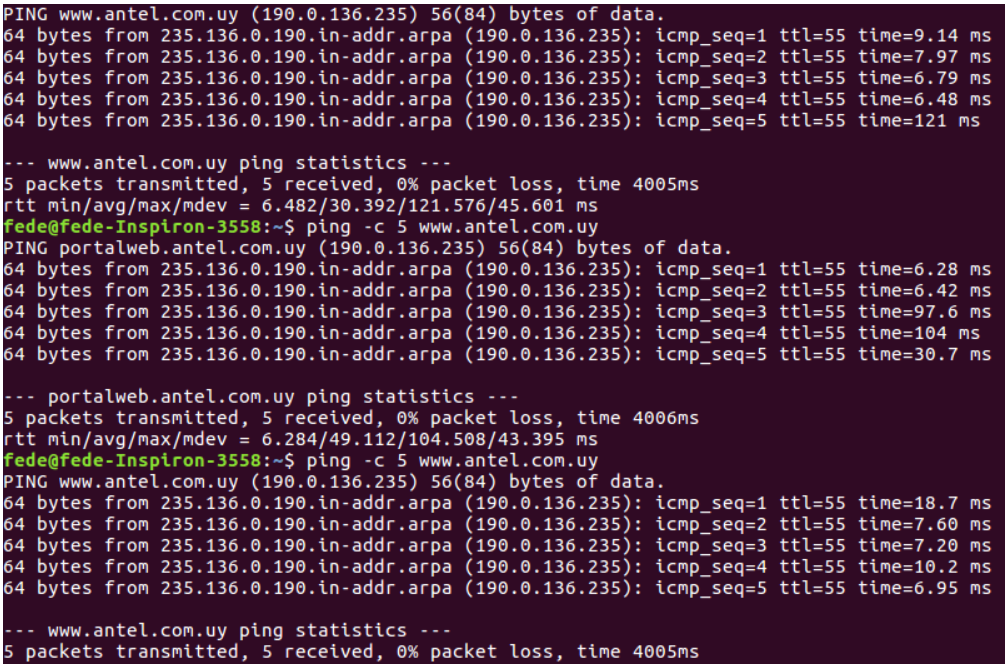

Figure 2: Impresión de la prueba.

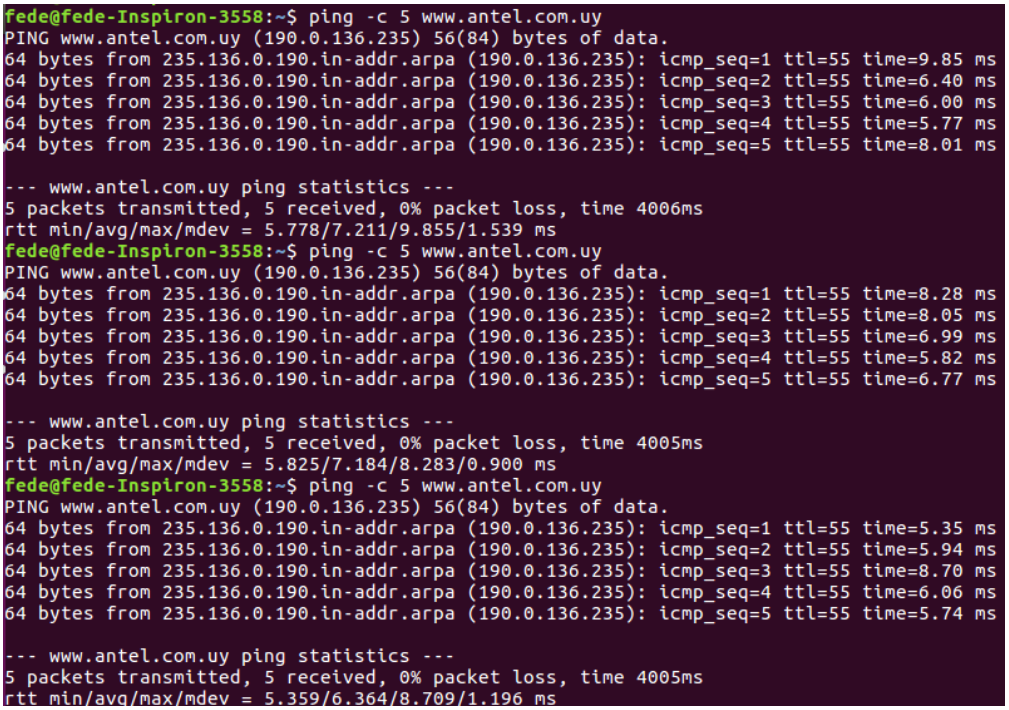

Figure 3: Impresión de la prueba.

1.3 Flag -n

1.3.1 Parte a

Ping a Amazon.com con la flag n (ping -c -n 10 www.amazon.com).
Con está flag la salida que se obtiene, sólo contiene valores numéricos, no se intenta buscar nombre simbólicos de las direcciones de los host. En este caso de prueba se puede observar que no se muestra los caracteres “server-52-84-173-227.gru50.r.cloudfront.net”.

1.3.2 Parte b

Esto puede ser útil al momento de analizar las salidas y solo visualizar los datos importantes tanto para el análisis humano como para la realización de un análisis computacional.

1.4 Tamaño de pruebas

1.4.1 Parte a

Dependiendo del sistema operativo que se esté utilizando es el valor predeterminado que tiene dicho campo:

Windows: 32 bytes
Linux: 64 bytes

El tamaño máximo que puede tomar este campo suele ser 65535 bytes.

1.4.2 Parte b

En las distintas pruebas a host como google.com, amazon.com y antel.com.uy se observaron distintos comportamientos.
En el caso de google se observa que los tamaños de los envíos son truncados a como máximo 72 bytes, con amazon se puede enviar pings de tamaños de 100, 1000 y 10000 sin problema y las respuestas son muy similares y en el caso de antel los envíos de tamaño 100 y 1000 se comportan normal con tiempos de respuesta similares pero al realizar el pedido de tamaño 10000 no responde y se pierden los paquetes. Esto creemos que puede suceder por dos razones, o bien se excede el tiempo de espera del ping o el host decidió no permitir la recepcion de paquetes tan grandes.
Creemos interesante resaltar que el tiempo de RTT no aumenta de forma lineal al crecer el tamaño de los paquetes.

2 Parte B - Captura de tráfico con Wireshark

2.1 Utilidad de la herramienta

Wireshark es un analizador de protocolos que permite capturar y analizar los paquetes de red que pasan por nuestro equipo. Permite ver en detalle todo el tráfico de una red y tener información y así poder solucionar o incluso prevenir posibles problemas de seguridad generado por conexiones ocultas que generan tráfico no

deseado a direcciones remotas. También es útil con fines académicos, permite el aprendizaje del funcionamiento de los diferentes protocolos de red ya que se puede observar de forma detallada las cabeceras de los protocolos estudiados, comprender la utilidad y función de cada uno de sus campos. La herramienta soporta más de 480 protocolos distintos y además cuenta con la posibilidad de examinar datos de una red viva o de un archivo de captura guardado en un disco duro. Cuenta con todas las características estándar de un analizador de protocolos.

Este programa nos permite tener la información detallada para poder analizar el tráfico que nuestra red tiene y en caso de encontrar algún problema solucionarlo o incluso prevenir futuros problemas.

2.2 Permiso de super-usuario

Los permisos de super-usuario son requeridos para realizar una captura de paquetes directa de la interfaz de red. Esto es debido a que existe un riesgo de seguridad del sistema al ejecutar los distintos analizadores de paquetes, ya que ante un bug del código en uno de ellos el sistema puede quedar expuesto a la ejecución de código externo malicioso. En nuevas versiones se implemento la separación de privilegios dentro del programa de forma de que se den privilegios únicamente al programa encargado de realizar la captura (Dumpcap) y así correr el resto de Wireshark con permisos normales.

2.3 Captura de tráfico

2.3.1 Parte a

En esta parte del análisis de la captura se aplico el filtro para los paquetes que utilizan el protocolo DNS. Es posible observar que se realizan consultas de tipo de registro A para la resolución de la IP según IPv4 Y AAAA para IPv6. En particular se observa entre las lineas 64 y 77 la resolucion DNS del dominio consultado mediante una respuesta de un registro CNAME. En dichas lineas se puede verificar que la dirección ip del servidor DNS consultado es 172.16.0.1.

2.3.2 Parte b

Para identificar una conexión TCP se busco el conjunto de 3 paquetes que se intercambian para establecer dicha conexión (Three-way handshake). Es decir que se debe encontrar un un envío SYN por parte del cliente al servidor, luego la respuesta del servidor mediante un SYN-ACK y finalmente el cliente envía un ACK al servidor para concretar la conexión entre ambos. Además se decidió filtrar por aquellos paquetes los cuales se le fue asignado como puerto destino el puerto 80 (por defecto para las conexiones tcp). En resumen el filtro aplicado fue: "tcp.flags.syn == 1 && tcp.flags.ack == 0 && tcp.dstport == 80" para poder contar solo los SYN y así estimar el numero de conexiones que se establecieron. Luego de aplicado el filtro los paquetes que cumplen esa condición son 12, en conclusión se establecieron ese número de conexiones TCP.

2.3.3 Parte c

Filtrando por el ptrotocolo HTTP, en la captura se observan varios pedidos HTTP al servidor mediante el método GET solicitando algún tipo de contenido. También se encuentran dichas respuestas del servidor informando que el pedido fue exitoso mediante un mensaje con código 200.

2.3.4 Parte d

Dentro de un mensaje de solicitud HTTP se encuentra la línea de cabecera User-agent que especifica el agente de usuario, es decir el tipo de navegador que esta haciendo la solicitud. Generalmente incluye información como el nombre de la aplicación, la versión, el sistema operativo, y el idioma. Los bots, a veces incluyen también una URL o una dirección de correo electrónico para que el administrador del sitio web pueda contactarse con el operador del mismo. Aplicando el filtro http.use_agent se puede observar en el detalle de los paquetes que el user-agent utilizado es Mozilla5.0.

2.3.5 Parte e

En cada sesión TCP se puede ver que ante los pedidos HTTP el servidor responde incluyedno información tal como la versión HTTP, un mensaje que indica el éxito

u error de la respuesta seguido de del tipo de contenido consultado por el cliente. Por ejemplo en las líneas 490 y 492 se obvserva el pedido y la respuesta por parte del servidor para cierto contenido consultado.

HTTP GET /searchbox-app/1.0.16/searchboxDrawer.js HTTP/1.1
HTTP/1.1 200 OK (application/javascript)

3 Parte C - Comando Traceroute

3.1 Funcionamiento del comando

Traceroute es una herramienta que determina la trayectoria que un paquete sigue para poder llegar a su destino, usando el parámetro TTL de los paquetes UDP o ICMP. Al ejecutar el comando traceroute nos devolverá la secuencia de saltos que atraviesa el paquete. Cada campo de vida útil perteneciente a los paquetes se van reduciendo en 1 cada vez que pasa por un router. En caso que el campo llegue a cero, el router detecta que hubo una circularidad, dando por finalizado este paquete y enviando una notificación ICMP al remitente. Por esta razón, Traceroute envía paquetes a un puerto UDP sin privilegios con su TTL configurado en 1. El primer router que se detecta eliminará el paquete y enviará un paquete ICMP adjuntando la dirección ip del router y la demora del bucle. Luego el traceroute va aumentando el TTL de a 1 hasta obtener una respuesta de cada router en la ruta, esto finaliza una vez que se obitene la respuesta "puerto ICMP Inalcanzable" de la máquina destino.

Es importante ver que el comando utiliza los protocolos ICMP (en windows) y UDP (Unix).

3.2 Ejecución del comando

3.2.1 Parte a

Al ejecutar traceroute a cada uno de los hosts del punto 2 de la parte A nos encontramos que algunos destinos no fueron alcanzados, estos son "www.antel.com.uy" y "zadna.org.za".

La ejecución de traceroute sobre los host que no fueron alcanzados se puede ver que aparecen asteriscos. Esto ocurre cuando los encaminadores que se cor-

responden con el salto indicado no responden con un mensaje indicando que el paquete ha sido descartado. En estos casos podemos ver un mensaje como el siguiente "Tiempo de espera agotado para esta solicitud". Puede deberse a que no se pueda establecer una conexión entre el punto de origen y el punto de destino elegido o que por un ahorro de tráfico los encaminadores no retornen un mensaje de error en los paquetes que se perdieron. También esto nos dice que el enrutador en ese salto no responde al protocolo mediante el cual se envian los paquetes para realizar Traceroute. Para los casos en que el destino no fue alcanzado, se ejecutó traceroute con un protocolo diferente ("traceroute -I www.antel.com.uy" forzando con esto el uso de paquetes ICMP en lugar de UDP que generaba problemas en determinados routers inermedio que no aceptaban paquetes de este protocolo y obtuvimos resultados satifactorios.

3.2.2 Parte b

Una vez que se ejecutó traceroute con el protcolo (ICMP) para los casos que no fueron alcanzados, comparamos todos los casos y obtuvimos que el host más proximo a nuestro punto de acceso en cantidad de saltos es antel y google (7 hops), esta respuesta coincide con lo analizado en la parte A mediante el uso del comando PING.

3.2.3 Parte c

Para las distintas ejecuciones del comando se obvservan que hay hops que responden con un tiempo mayor que otro dentro del mismo recorrido, es posible que el router asociado a ese host tenga un trabajo mayor de procesamiento de paquetes que el resto o se deba a la situación actual de la red asociada a dicho host. Estos routers se encuentran en distintas partes del mundo, por lo cual hay algunos saltos claves que marcan la diferencia de tiempo en la llegada del paquete a un sistema final.

3.2.4 Script Traceroute

Para la implementación del script se utilizó como base el algoritmo sugerido, en primer lugar se analizó el mismo para comprender dicho funcionamiento. En el

analisis observamos que se utiliza un socket el cual conecta dos programas ubicados posiblemente en dos computadoras distintas por el cual pueden intercambiar cualquier flujo de datos de manera fiable y ordenada. En este caso cumple el funcionamiento para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto. Se recibe como primer parámetro el host destino, el segundo es la cantidad de bytes que se le asigna al tamaño del payload del paquete UDP. En tercer lugar se pasa la flag que indica si mostrar o no los nombres de los dominios asociados a cada hop(Si la flag esta en V entonces solo se muestran las IP). En ultimo lugar se pasa por parámetro un nombre de o dirección asociada a un dominio y se indica si coincide con alguno de los hops.

4 Parte D - Comando Dig

4.1 Sistema DNS

Existen dos formas posibles para identificar un host, siendo estas el nombre del mismo y su dirección IP. Generalmente, los usuarios no prefieren trabajar con las IP debido a la dificultad de recordar dicho numero, sino que prefieren trabajar con un nombre dado. Además, puesto que los nombres de host pueden constar de caracteres alfanuméricos de longitud variable, podrían ser difíciles de procesar por los routers. Por estas razones, los hosts se identifican mediante direcciones IP y además también brinda más información, como de ser la ubicación del host. Para relacionar ambos identificadores se cuenta con un mecanismo que permite traducir nombres en lenguaje natural a direcciones numéricas este el sistema DNS. El sistema de nombres de dominio (DNS, por sus siglas en inglés, Domain Name System) es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada. Este sistema cumple la función de mapear las direcciones IP asociados con los equipos conectados a la red con sus dominios, cumpliendo la función de facilitar la conexión entre los equipos ya que la dirección IP puede variar en el tiempo y no es mnemotécnica.

La base de datos distribuida implementada jerárquicamente como un árbol, cada nodo u hoja del árbol tiene cero o más registros de recursos asociados, los

cuales a su vez contienen información asociada al nombre del dominio. La base de datos se divide en secciones llamadas zonas, las cuales son sub árboles del DNS y se administran de forma separada. Los name servers asociados a una zona en concreto son los responsables de responder las consultas hechas en dicho lugar. Pueden existir varios name servers asociados a la misma, aunque usualmente uno es el primario y los otros secundarios, un name server puede ser autoritativo para más de una zona. Es común que varios usuarios que no se encuentran familiarizados con el concepto de DNS suelen creer que no tienen conexión a Internet cuando en verdad están experimentando un problema con el DNS. Esto se debe a que los servidores DNS viven en Internet y el dispositivo de cada usuario los contacta para que pueda realizar la traducción nombre-IP. Por lo cual, si hay algún retraso en conectarse al servidor DNS ,o si el servidor demora mucho tiempo en poder resolver la IP, causará un retraso en el ingreso al sitio. Esto conlleva que incluso teniendo la conexión mas rápida de todas la navegación se note lenta.

4.2 Servidor Recursivo/Iterativo

Cuando se realiza un lookup existen dos formas posibles de resolución, estas son de forma iterativo o de forma recursiva. En la forma iterativa el host realiza la consulta al servidor local de DNS, y es este el encargado de consultar a los distintos servidores correspondientes de manera tal que cada uno busca la dirección IP en su caché y la retorna, en caso de no tenerla le indica al servidor local a quien realizar la siguiente consulta, y obtener así la dirección IP solicitada. De la otra forma, en la resolución recursiva luego de que el host realiza la consulta al servidor de DNS local, este transmite dicha consulta al Servidor raíz correspondiente y espera a obtener como respuesta la dirección IP final ya resuelta. Cada uno de los servidores DNS en la recursión repite este mismo proceso hasta obtener la IP requerida o llegar al Servidor Autoritativo correspondiente que sea capaz de resolver la consulta. Otra diferencia existente es que en la resolución recursiva los servidores se cargan con mayor trabajo ya que deben mantener los estados de las request que aún no se han respondido para saber a donde enviar la respuesta, mientras que en las consultas iterativas la carga cae en el servidor local.

Comando DIG para consulta del registro A del dominio redhat.com.

En primer lugar cuando se consulta al servidor 8.8.8.8 se obvserva que el tiempo de consulta es notoriamente menor con respecto ala otra consulta. (Query time: 47msec en lugar de 587msec). También se observa que consulta al servidor 192.42.93.30 se realiza de forma iterativa dado que no es posible realizarla recursivamente con dicho servidor, En cambio la consulta utilizando 8.8.8.8 se realiza en forma recursiva. Investigando sobre este servidor DNS de uso público se puede saber que es brindado por google y tiene como objetivo brindar un acceso más rápido a internet. Es posible hacer uso de este servicio DNS en lugar del previsto por defecto por el proveedor de internet. Dada la infraestructura de dicho DNS permite que la navegación sea más rápida y se permita el acceso a contenido que haya sido bloqueado por el proveedor de internet. En el caso de este último se concluye que la consulta es resuelta mediante un registro del tipo CNAME.

4.3 Script Dig 1

Para la resolución de este problema creamos un script basado en lenguaje python donde se importaron las clases subprocess, reg y sys. En este nos basamos en generar consultas con dig y el análisis de estas utilizando expresiones regulares. Al comienzo del script se compilan estas expresiones regulares. En este se genera una consulta dig hacia la dirección del root server 192.203.230.10 con un dominio pasado por parámetro, del resultado que se obtiene se toman las ip del registro A y a partir de esas ip se resuelve nuevamente la misma consulta hasta que se obtenga la ip a buscar. En caso de no haber registro A se busca un CNAME y a partir de este se busca su ip basándonos en 192.203.230.10 y a partir de esta se comienza de nuevo. En caso de no haber CNAME nos basamos en los registros NS de la búsqueda anterior. Para comprobar el funcionamiento del script se realizó una captura de tráfico con Wireshark durante la ejecución del script Iterativo.py para resolver el host www.samsung.com. Analizando la captura se observa en primer lugar la consulta desde el host local al root server elegido, se ven las consultas por los dos tipos de registro A y AA, seguido de esto se tiene la respuesta del root server la cual contiene una lista nameservers intermedios. A continuación se tiene la misma cosulta inicial pero esta vez con destino al servidor 192.5.6.30 el cual fue obtenido de la consulta anterior. Una vez mas se obtiene como respuesta

una lista nameservers esta vez son menos y se realiza la consulta a uno de ellos nuevamente(dnssm.samsung.com). Más abajo se tiene como respuesta un registro de tipo CNAME, indicando que www.samsung.com.akadns.net es un alias de el dominio que estamos intenando resolver. Luego de esto se realiza una consulta por dicho alias contra el servidor root inicial, de la mismo forma se aprecian las consultas sobre este alias hasta que finalmente se obtiene por parte del servidor 192.168.1.43 una respuesta de registro tipo A con la ip del dominio buscado. De lo obtenido se puede deducir que la consulta se realiza de forma iterativa, ya que desde el host local se realizan todas las consultas necesarias hasta recibir la respuesta del servidor autoritativo del dominio consultado con su IP.

4.4 Script Dig 2

En este script se busca conseguir la dirección ip de el servidor del dominio de correos electrónicos de el dominio pasado por parámetro. Para esto utilizando las mismas herramientas y lenguaje que el script anterior, se realiza una consulta dig con las flags MX, noall y answer con el dominio pasado por parametro, para obtener los dominios de los servidores del mail del dominio. Estas flags nos ayudan a obtener el registro MX (registro de dominio de mail) y para que estos sean servidores autoritativos de ese dominio. Con los dominios obtenidos se genera una consulta dig con la flag short para que únicamente responda con la dirección ip de el dominio este resultado se compara con la ip pasada por parámetro y en caso de que alguna de las comparaciones sean iguales se retorna True.

4.5 Consulta DNS inversa

Para obtener el nombre de un host dada la direccion IP se debe ejecutar el comando dig junto con la flag x es decir: dig -x 200.40.30.218. La respuesta que se obtiene ejecutando el comando con dicha ip es el servidor de mail adinet. El funcionamiento del comando con esta flag encendida se basa en la ejecución de una consulta al registro PTR, este registro es el opuesto al A. Es decir que que un registro PTR resuelve una IP a que apunta a un Dominio o servidor.

References

[Man Ping Linux] https://linux.die.net/man/8/ping

[Ping Wikipedia] https://es.wikipedia.org/wiki/Ping

[Laura Chappell, Sr. Protocol Analyst] ftp://ftp.hp.com/pub/hpcp/UDP-ICMP-Traceroutes.pdf

[Wireshark] http://www.wireshark.org/docs/wsug html chunked/

[Wireshark Wikipedia] https://wiki.wireshark.org/CaptureSetup/CapturePrivileges

[Traceroute] http://www.blog.gnutic.com/2015/09/que-es-y-como-funciona-el-comando-traceroute/

[Dig] https://es.wikipedia.org/wiki/Sistema_de_nombres_de_dominio

[Dns] http://www.ehowenespanol.com/consulta-recursiva-vs-iterativa-dns-info_232082/

[James F. Kourose] James F. Kourose and Keith W. Ross. Redes de computadoras: un enfoque descendente, 5.a edición

[Dns] https://es.wikipedia.org/wiki/Socket_de_Internet