Trabajo Práctico - DNS Solver

Tecnología Digital IV: Redes de Computadoras
Licenciatura en Tecnología Digital
Primer Semestre, 2024

Bautista Lobo, Juan Ignacio Elosegui, Massimo Giammaria

Viernes 14 de junio, 2024

Introducción

DNS es el sistema de nombres de dominio -*Domain Name System*-, que relaciona nombres legibles por seres humanos con direcciones IP. Es un protocolo de aplicación y ofrece la infraestructura para implementar esto de manera eficiente.

Cuando un usuario ingresa un nombre de dominio en un navegador web o cualquier otra aplicación que requiera una resolución de nombres, el sistema operativo del dispositivo realiza una consulta DNS para traducir el nombre de dominio en una dirección IP.

Un *DNS Solver* es un programa que resuelve consultas DNS. Hace de intermediario entre un cliente que realiza una consulta de resolución de nombres de dominio y los servidores DNS que contienen la información necesaria para responder a esa consulta.

El objetivo de este trabajo es poder implementar un DNS Solver implementado en Python que resuelva consultas DNS de tipo A de forma iterativa, consultando a toda la jerarquía de servidores DNS.

1 Primera Parte: Implementación del DNS Solver

Ejercicio 1.1

En este ejercicio, se implementó un DNS Solver básico que realiza consultas DNS de tipo A de forma iterativa. A diferencia de la versión mejorada del ejercicio 1.2, en este caso, el programa elige un servidor root aleatorio para realizar la primera consulta DNS.

1.1.1 - Imports y Módulos

Al igual que en el ejercicio 1.2, utilizamos las librerías **Scapy**, **Socket**, y **Time**. Scapy nos permite manipular paquetes, Socket proporciona la interfaz de red de bajo nivel, y Time se usa para medir los tiempos de respuesta.

1.1.2 - Configuraciones

Para esta implementación, hemos configurado una lista de servidores root y el programa selecciona uno aleatoriamente para iniciar la consulta. La lista de servidores root disponibles es la misma que en el ejercicio 1.2:

- 198.41.0.4
- 199.9.14.201
- 192.33.4.12
- 199.7.91.13
- 192.203.230.10
- 192.5.5.241
- 192.112.36.4
- 198.97.190.53
- 192.36.148.17
- 192.58.128.30
- 193.0.14.129
- 199.7.83.42
- 202.12.27.33

1.1.3 - Construcción de la Consulta

La construcción de la consulta es idéntica a la descrita en el ejercicio 1.2. Usamos la clase DNS de Scapy para crear una consulta de tipo "A", con el parámetro `rd` en 0 para que el servidor DNS no permita la recursión.

1.1.4 - Envío de la Consulta

El envío de la consulta también sigue el mismo procedimiento que en el ejercicio 1.2. Configuramos un *socket* UDP para enviar la consulta DNS al servidor root seleccionado aleatoriamente, a través de puerto 53 -de DNS-, con un timeout de 2 RTT (*Round Trip Time*), que representa el tiempo de ida y vuelta de la consulta.

1.1.5 - Resolución de la Consulta

Para resolver la consulta, seguimos un procedimiento iterativo similar al del ejercicio 1.2. La única diferencia es que el servidor root inicial se selecciona aleatoriamente en lugar de ser elegido por el usuario.

El flujo de trabajo para resolver la consulta es el siguiente:

- Elegir un servidor root aleatorio de la lista.
- Enviar la consulta DNS al servidor root.
- Procesa la respuesta recibida, revisando el tipo ya sea CNAME, NS o A, para luego bajar en la jerarquía de forma iterativa usando un algoritmo recursivo que toma como parámetro los dominios de las capas a consultar, además de usar sets para evitar reconsultar dominios ya visitados.
- Si la respuesta contiene direcciones IP, devuelve las IPs resueltas o revisa los campos adicionales en busca de otras posibles respuestas.
- Si la respuesta contiene servidores de autoridad, enviar consultas iterativas a estos servidores hasta obtener las direcciones IP del dominio consultado, si no se obtiene ninguna respuesta el código informa que no hubo respuesta.

2 Segunda Parte: Experimentación

2.1 Primer Experimento: YouTube al Mediodía y a la Medianoche

2.1.1 - Introducción

Con este experimento queríamos ver qué pasaba con los tiempos de respuesta de las consultas DNS cuando queríamos acceder a la IP de un dominio en horarios en los cuales sabemos que reciben más tráfico.

Pensamos en YouTube (`www.youtube.com`), que sabemos con certeza que se recibe mucho más tráfico cerca de la medianoche, ya que la gente consume contenidos mientras cena, para poner de fondo, o para ver antes de dormir. También asumimos que durante el mediodía no recibe esta página tanto tráfico, porque la gente trabaja, está en la escuela, en la universidad, o durmiendo la siesta.

2.1.2 - Implementación

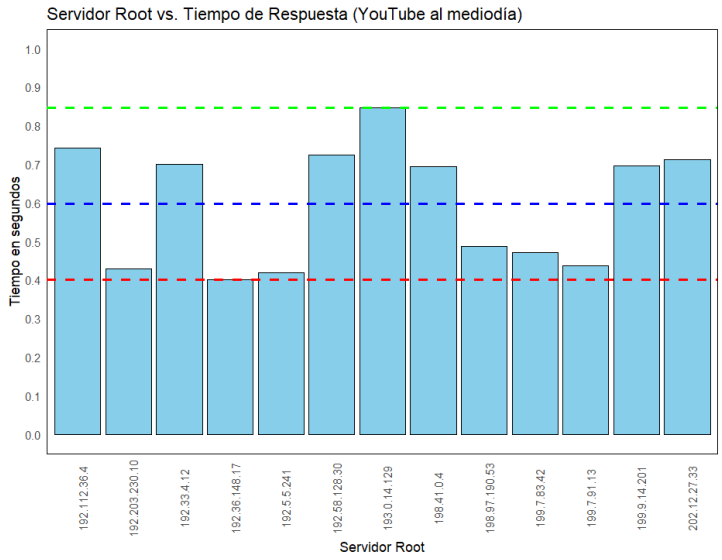
Para recolectar las métricas, probamos todos los servidores root (ver lista en la sección 1.2.2) y analizamos los tiempos de respuesta cuando quisimos consultar por el dominio de YouTube. Hicimos esto al mediodía y a la medianoche.

Los tiempos los guardamos en archivos llamados `yt_mediodia.csv` y `yt_mediodia.csv`.

Luego, calculamos los tiempos medios de respuesta con el software *R*, hicimos gráficos de barras con los tiempos de respuesta por cada dirección IP del servidor DNS al cual le hicimos la consulta.

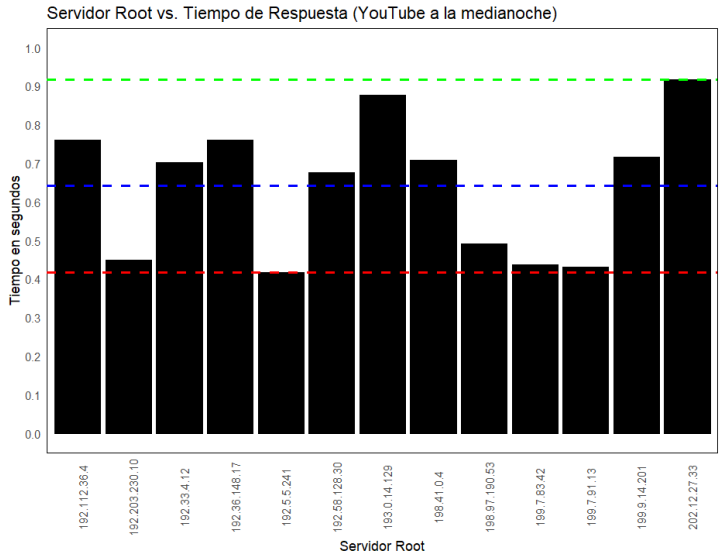
2.1.3 - Resultados

Veamos qué sucedió cuando hicimos 15 consultas a 15 servidores DNS distintos al mediodía buscando la IP de `www.youtube.com`:



- Se puede ver que el tiempo de respuesta más rápido fue con el servidor DNS `192.36.148.17`, con 0.4012 segundos. Caracterizado con la línea punteada roja.
- El tiempo de respuesta más lento fue el que corresponde al servidor DNS con dirección IP `193.0.14.129`, con 0.8458 segundos. Caracterizado con la línea punteada verde.
- El tiempo medio que tardaron los servidores DNS en brindarnos la IP fue de 0.5969 segundos. Caracterizado con la línea punteada azul.

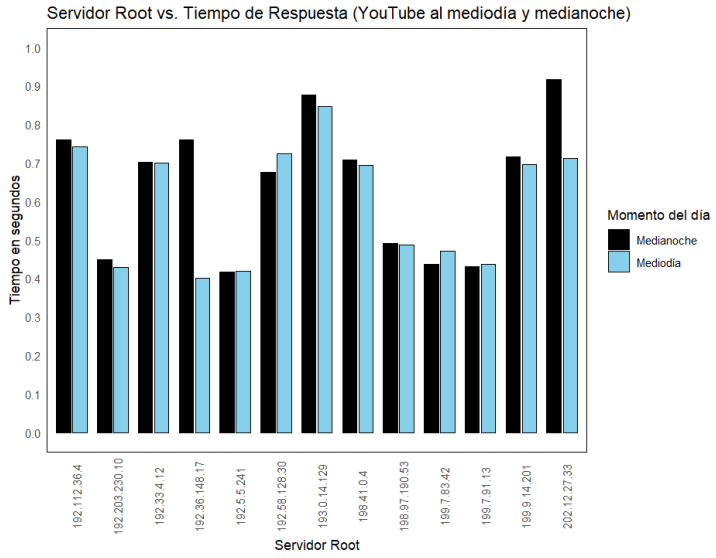
Ahora veamos qué sucedió cuando hicimos las mismas 15 consultas a los 15 servidores distintos:



- Se puede ver que el tiempo de respuesta más rápido fue con el servidor DNS `192.5.5.241`, con 0.4169 segundos. Caracterizado con la línea punteada roja.
- El tiempo de respuesta más lento fue el que corresponde al servidor DNS con dirección IP `202.12.27.33`, con 0.9174 segundos. Caracterizado con la línea punteada verde.
- El tiempo medio que tardaron los servidores DNS en brindarnos la IP fue de 0.6417 segundos. Caracterizado con la línea punteada azul.

2.1.4 - Conclusiones

Para comparar de manera correcta los dos resultados, construimos el siguiente gráfico:



- Existe una variabilidad en los tiempos de respuesta de los servidores DNS tanto al mediodía como a la medianoche. Sin embargo, los tiempos de respuesta son en promedio más altos a la medianoche.
- Los resultados muestran que los tiempos de respuesta de los servidores DNS son generalmente más lentos a la medianoche en comparación con el mediodía. Esto es coherente con nuestra hipótesis de que el tráfico es mayor en este horario, lo que puede causar una mayor carga en los servidores DNS y, por lo tanto, tiempos de respuesta más largos.

- En ambos horarios, hay servidores que destacan por ser consistentemente más rápidos o más lentos. Esto sugiere que además del tráfico, otros factores como la ubicación geográfica del servidor, su capacidad y su carga actual también pueden influir significativamente en el tiempo de respuesta.

En resumen, el experimento confirma nuestra hipótesis inicial: los servidores DNS tardan más en responder a la medianoche, probablemente debido a un mayor tráfico en ese horario. Esto destaca la importancia de considerar el momento del día y la carga del servidor al evaluar los tiempos de respuesta de las consultas DNS.

2.2 Segundo Experimento: Relación Lineal Distancia-Tiempo de Respuesta

2.2.1 - Introducción

Con el grupo pensamos que existía una relación lineal entre la distancia a un país cualquiera con el tiempo de respuesta de un dominio local de ese país. Esto es justamente lo que queremos averiguar: ¿la distancia hacia otro país nos asegura un mayor tiempo de respuesta a la hora de conseguir la IP de un dominio de ese lugar?

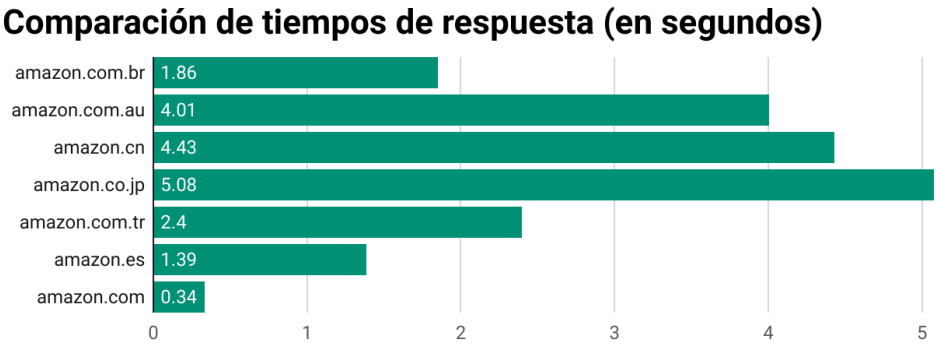
2.2.2 - Implementación

Nosotros probamos esto con Amazon. Lo que hicimos fue conseguir la IP a través del DNS Solver con un servidor root aleatorio de los siguientes países:

- Brasil [`amazon.com.br`] (2800km de distancia)
- Australia [`amazon.com.au`] (13.226km de distancia)
- China [`amazon.cn`] (30.675km de distancia)
- Japón [`amazon.co.jp`] (20.000km de distancia)
- Turquía [`amazon.com.tr`] (20.000km de distancia)
- España [`amazon.es`] (10.000km de distancia)
- Estados Unidos [`amazon.com`] (8.000km de distancia)

2.2.3 - Resultados

Luego de recolectar los tiempos, los volcamos todos en un gráfico que hicimos con el software de *Flourish*. Es importante notar que el eje X está ordenado por distancia desde Buenos Aires hacia aquel país.



2.2.4 - Conclusiones

Se puede ver que claramente no existe tal relación lineal entre la distancia del país y el tiempo de respuesta del servidor DNS para ofrecernos la dirección IP del dominio de Amazon local. Se puede ver el contraejemplo de manera clara cuando comparamos los tiempos y distancias entre Australia y Turquía: Turquía está 7.000km más lejos que Australia, aún así se pudo conseguir la IP de `amazon.com.tr` 1.6 segundos más rápido. ¿Por qué pasa esto? Según lo que averiguamos, se puede dar por varios factores:

- La calidad y capacidad de la infraestructura de red varía significativamente entre países. Algunos países tienen redes más avanzadas y mejor mantenidas, lo que puede reducir los tiempos de respuesta, independientemente de la distancia.

- Los servidores DNS pueden utilizar cachés para almacenar temporalmente las respuestas a consultas recientes. Esto puede acelerar el tiempo de respuesta para dominios que se han consultado recientemente, sin importar la distancia geográfica.
- Los proveedores de servicios, como Amazon, optimizan sus servidores para mejorar el rendimiento global. Esto incluye el uso de CDN (Content Delivery Networks) y servidores distribuidos geográficamente para reducir el tiempo de acceso a sus servicios.
- El tráfico de datos no siempre toma una ruta directa entre el origen y el destino. Los datos pueden pasar a través de varios intermediarios, lo que puede aumentar el tiempo de respuesta debido a la necesidad de más saltos.