

# Sistema de Nombres de Dominio (DNS)

## Laboratorio 01 (L01)

Luis Golac, Sebastian Loza, Paolo Vásquez

Redes y Comunicaciones (CS4054) - Laboratorio 2.01 - 2025 - 1

UTEC - Universidad de Ingeniería y Tecnología, Lima-Perú

### I. INTRODUCIÓN

Actualmente estamos, por no decir siempre, enviando, recibiendo y solicitando información; es decir, constantemente estamos comunicándonos. Para una correcta comunicación, normalmente se considera únicamente necesario tener claro con quién se desea comunicar. Sin embargo, también es requerido saber a dónde comunicar; es decir, dónde se encuentra con quien deseamos comunicar. A un nivel más alto, por ejemplo, de redes o sistemas, esto también sucede. El modelo de interconexión de sistemas abiertos (OSI, por sus siglas en inglés), el cual detallaremos un poco más adelante, define básicamente cómo se comunican los sistemas de redes; cómo hacen consultas, envían datos y también los reciben. Este modelo define diferentes capas o niveles, dentro de las cuales aparece la capa de aplicación, el foco de este estudio. A través de esta experiencia, dentro de la capa de aplicación, exploraremos el sistema de nombres de dominio (DNS), comprendiendo cómo es posible obtener direcciones o ubicaciones a partir de nombres de dominio. Esto nos permitirá entender el mecanismo mediante el cual se determina dónde comunicar, es decir, la ubicación de la entidad con la que se desea establecer una comunicación. También, aunque no es el foco de este estudio, aparecerá en algunas secciones la capa de transporte, con su protocolo de transmisión (TCP).

#### I-A. Marco Tórico

Previo a la experiencia, es necesario tener claras ciertas nociones teóricas. A manera de síntesis, revisaremos los siguientes contenidos: **Capa de Aplicación, Sistema de Nombres de Dominio (DNS), Name Server Lookup, Ipconfig y similares, Capa de Transporte y TCP, Wireshark y Adicionales.**

##### 1. Capa de Aplicación

El modelo OSI [1], como se detalló anteriormente, define ciertas capas, específicamente siete, que se apilan de abajo hacia arriba. En orden de menor a mayor, estas capas son: física, enlace de datos, red, transporte, sesión, presentación y aplicación. En el

nivel más alto aparece la capa de aplicación [2], la cual se comunica directamente con el usuario. Aunque una aplicación puede interactuar con otras capas del modelo OSI, finalmente la interfaz se ejecuta en esta capa. Cuando un usuario promedio recibe un mensaje en cierto software es esta capa la que se lo presenta. Dentro de esta, uno de los protocolos más recurrentes es el DNS, que veremos a continuación. Note que para lo siguiente, es fundamental entender los conceptos de IP [3] (del inglés *Internet Protocol*), una dirección numérica que identifica de manera única a un dispositivo, y dominio [4], nombre textual que se usa para identificar una dirección IP. Una IP se ve como 192.168.1.1, mientras que un dominio como utec.edu.pe.

##### 2. Sistema de Nombres de Dominio (DNS)

El sistema de nombres de dominio (DNS) [5], a grandes rasgos, traduce los nombres de dominio a direcciones IP aptas para la lectura por parte de las máquinas (aunque veremos que no solo permite hacer esto), y es por ello que, al buscar dominios en el navegador, podemos acceder a los recursos asociados.

DNS es, básicamente, una base de datos jerárquica [6], compuesta por tres niveles principales: *Root*, *TLD (Top-Level Domain)* y *Authoritative* (Figura 1). *Root* contiene la información de los servidores TLD; luego, en TLD se almacenan los dominios de nivel superior (.com, .org, .net); y finalmente, *Authoritative* contiene la información real del dominio. Por ejemplo, al hacer una búsqueda de www.google.com, *Root* nos indicaría revisar los servidores TLD de .com; luego, TLD nos enviaría a los servidores autoritarios (*authoritative*) de Google; y finalmente, estos nos dirían que la IP de www.google.com es 142.250.190.4. Para hacer estas búsquedas, existen dos métodos: iterativo (Figura 2) y recursivo (Figura 3) [6]. En el método iterativo, desde el servidor local se consulta el servidor *Root*, luego se vuelve al local, y así sucesivamente hasta llegar al servidor *Authoritative*.

En cambio, en el método recursivo, la consulta va desde el servidor local hacia *Root*, luego al TLD y, finalmente, al *Authoritative*, regresando por el mismo camino hasta el servidor local con la respuesta.

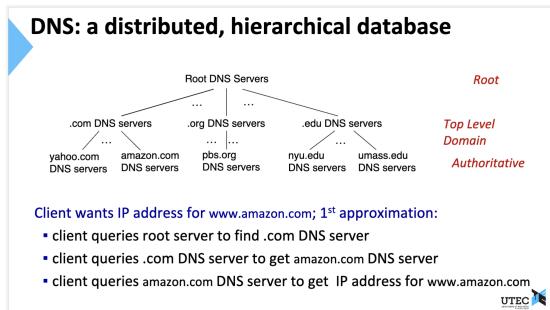


Figura 1: Estructura de DNS. Se constituye de los niveles Root, Top Level Domain y Authoritative y la búsqueda se realiza descendientemente, desde Root hasta Authoritative [6].

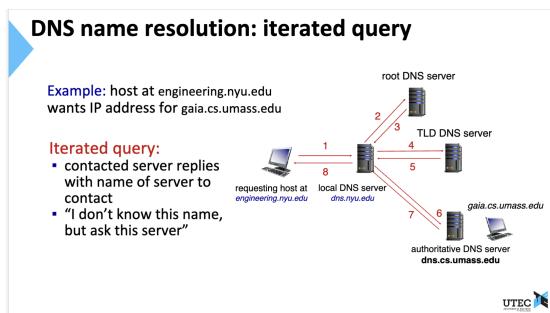


Figura 2: Consulta de tipo iterativa para DNS. Iterativamente se consulta un servidor externo y se vuelve al local, hasta alcanzar al authoritative, y de ahí se consigue la respuesta [6].

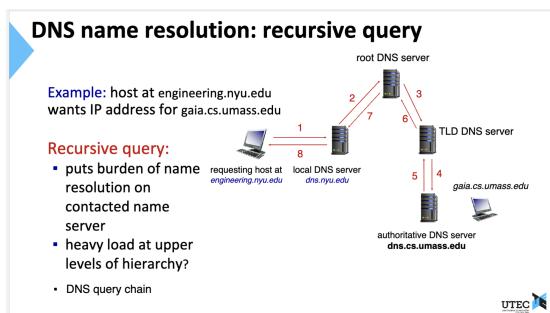


Figura 3: Consulta de tipo recursivo para DNS. Recursivamente se abre un camino hasta un servidor authoritative, y se vuelve de retroceso en el camino con la respuesta [6].

DNS puede almacenar en *cache* búsquedas anteriores, a manera de simplificar búsquedas repetidas, y lo hace en el formato de base de datos distribuida que almacena registros de recursos (*Resource Records, RR*) [6] (Figura 4). Así, DNS no solo realiza consultas de direcciones IP, sino también de alias, nombres de servidor y servidores

de correo asociados a un dominio. Todos estos datos se almacenan según el formato antes descrito. Además, los mensajes DNS poseen un formato específico [6], importante de conocer y entender (Figura 5). Primero, constan de una identificación (*identification*) y ciertos parámetros (*flags*). Luego, se incluyen la consulta específica del dominio (*questions*), las respuestas a la consulta (*answers*), que pueden ser de tipo A, NX, CNAME o MX, la información sobre qué servidor tiene autoridad sobre el dominio consultado (*authority*) y cierta información adicional (*additional info*). Finalmente, los campos marcados con “#” indican la cantidad de elementos en sus respectivos campos.

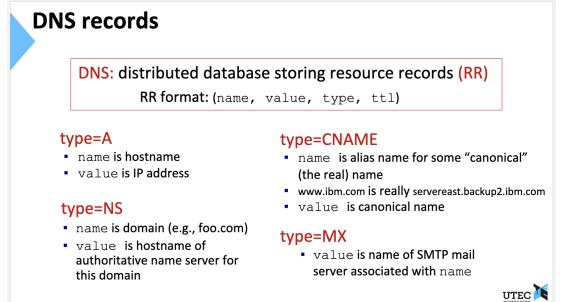


Figura 4: Estructura de resource records para DNS. Se define el formato (name, value, type, ttl) y se muestran los cuatro tipos de respuesta comunes: A, CNAME, NS y MX [6].

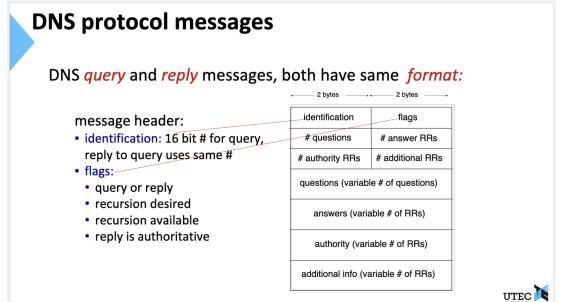


Figura 5: Formato de mensaje DNS. Se constituye del header (identification y flags), los parámetros pedidos y recibidos (question, ..., additional info) y la cantidad de data transmitida (questions, ..., additional RRs) [6].

### 3. Name Server Lookup

Name Server Lookup (nslookup) [7] es básicamente una herramienta de línea de comandos que se utiliza para consultar servidores DNS y ver información sobre nombres de dominio, direcciones IP y otros aspectos relacionados. Podemos consultar registros específicos de DNS con nslookup -type={type} {dominio}, donde {type} puede ser, siguiendo lo anterior visto, A, MX, NS, CNAME, entre otros, y {dominio} el nombre del dominio a consultar. Por defecto, el tipo es A.

Se pueden realizar consultas empleando servidores DNS específicos, como por ejemplo con nslookup google.com 8.8.8.8; sin embargo, si esta IP no es una dirección de servidor DNS válida, la consulta no será satisfactoria.

También es posible hacer consultas inversas, obteniendo el nombre de dominio a través de la IP, con por ejemplo nslookup 8.8.8.8 (note nuevamente que si es inválida no procederá). Entonces, a manera de ejemplo, si realizamos nslookup google.com (usa por defecto el tipo A, es decir IP), obtenemos el resultado de la Figura ???. El primer Address mostrado es el del servidor empleado para realizar la consulta, y luego los demás Address con sus respectivos Name, son básicamente las direcciones IP donde se encuentra el dominio consultado.

```

paolovasquezgrahammer -- zsh -- 80x24
Last login: Fri Apr 25 17:39:42 on ttys028
paolovasquezgrahammer@Paolos-MacBook-Pro ~ % nslookup google.com
Server: 200.48.225.130
Address: 200.48.225.130#53

Non-authoritative answer:
Name: google.com
Address: 108.177.123.139
Name: google.com
Address: 108.177.123.101
Name: google.com
Address: 108.177.123.102
Name: google.com
Address: 108.177.123.113
Name: google.com
Address: 108.177.123.100
Name: google.com
Address: 108.177.123.138
paolovasquezgrahammer@Paolos-MacBook-Pro ~ %

```

Figura 6: Consulta default con nslookup

#### 4. Ipconfig y similares

Ipconfig [8] es una herramienta que muestra todos los valores actuales de configuración de red TCP/IP, del sistema de nombres de dominio (DNS), entre otros. De manera simple, ipconfig (sin ningún otro parámetro) muestra las direcciones IP, la máscara de subred y la puerta de enlace predeterminada. Note que la **máscara de subred** [9] define cuántas direcciones IP pertenecen a una misma red, y la **puerta de enlace predeterminada** [10] es la dirección IP del router que conecta una red local con otras redes.

Los principales parámetros de ipconfig (y los que se usarán en la experiencia), son /all, que muestra la configuración completa de TCP/IP, /displaydns, que muestra el contenido de la caché del solucionador de cliente DNS y /flushdns, que vacía y restablece el contenido de la caché de resolución del cliente DNS. Cabe mencionar que esto funciona meramente para el sistema operativo Windows, y se deja un pequeño resumen para macOS y Linux [11]:

ipconfig /all

**macOS/Linux:** ifconfig -a

ipconfig /displaydns

<b>macOS:</b>	sudo killall -INFO mDNSResponder
<b>Linux:</b>	systemd-resolve -statistics

ipconfig /flushdns

<b>macOS:</b>	sudo dscacheutil -flushcache; sudo killall -HUP mDNSResponder
<b>Linux:</b>	sudo systemd-resolve -flush-caches

A manera de ejemplo, al ejecutar ifconfig -a (en macOS), obtenemos lo mostrado en la Figura 7, siendo en0 la máscara de subred.

```

paolovasquezgrahammer -- zsh -- 80x24
nd6 options=201<PERFORMNUD,DAD>
media: <unknown type>
status: inactive
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=6460<TS04,TS06,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
ether f6:00:d1:30:75:04
nd6 options=201<PERFORMNUD,DAD>
media: autoselect (None)
status: inactive
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=6460<TS04,TS06,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
ether 1a:fe:31:60:d7:b2
inet6 fe80::b5:1a61:500f:1211en0 prefixlen 64 secured scopeid 0xe
inet6 2001:1388:80c:d55e:c05:8746:f1e5:ab57 prefixlen 64 autoconf secure
d
y
inet6 2001:1388:80c:d55e:5ae:d3d3:66:ab83 prefixlen 64 autoconf temporar
y
inet 192.168.1.82 netmask 0xffffffff broadcast 192.168.1.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
awdl0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=6460<TS04,TS06,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSUM>
ether 8e:b7:48:08:86:c2

```

Figura 7: Ejecución ifconfig -a en macOS

#### 5. Capa de Transporte y TCP

De forma breve, porque no es el foco de esta experiencia, esta capa, la número cuatro [2], se encarga de tomar los datos y dividirlos en partes más pequeñas; ello para poder transferirlos de un lugar a otro. Dentro de esta capa, tenemos el protocolo de control de transmisión [12] (TCP), el cual permite intercambiar mensajes a través de una red. HTTP utiliza TCP por debajo para poder transferir mensajes. Es importante tener noción de lo que son los segmentos TCP; se pueden entender como la cantidad de partes en las que TCP dividió un mensaje para transmitirlo, y en muchos casos, esa cantidad para un mensaje específico dependerá

de la tarjeta de red del dispositivo empleado.

## 6. Wireshark

Este software es un analizador de paquetes de código abierto [13]. Básicamente, nos permite analizar la red de manera muy minuciosa, dándonos herramientas y comandos para filtrar y examinar con detalle el tráfico de la red. A manera de ejemplo, nos podría permitir observar paquetes caídos o incluso actividad maliciosa en alguna red. También, y de forma general, permite aprender sobre protocolos como HTTP y TCP, analizar y comprender los encabezados de paquetes, su enrutamiento y demás aspectos.

Sobre su utilización [14], de manera breve: al instalar la aplicación podrá ver el entorno que se observa en la Figura 8. Para iniciar la captura de paquetes, seleccione la alternativa que desee (pruebe con Wi-Fi: en0) y podrá ver cómo se capturan paquetes como en la Figura 9. En esta última podrá observar las partes relevantes. Para detener la captura, presione el botón cuadrado rojo, y para iniciar nuevamente la captura, presione el botón azul con forma de aleta de tiburón.

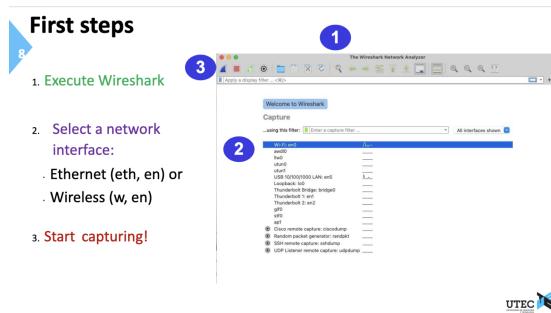


Figura 8: Indicaciones generales para usar Wireshark. 1 se debe ejecutar Wireshark, 2 seleccionar alguna interfaz (por ejemplo Wi-Fi) y 3 comenzar la captura con el botón azul [14]

En esta experiencia, como ya debe suponerse, haremos uso de Wireshark para monitorear la comunicación entre nosotros (cliente) y diferentes servicios (servidores), con ello viendo y entendiendo de manera práctica un poco de lo explicado anteriormente. Se comprenderá también un poco más la utilidad de este software, aunque también veremos más adelante cómo se ha usado en diferentes estudios.

## 7. Adicionales

A manera adicional, y por si todavía no está claro, el propósito de DNS es convertir nombres de dominios a direcciones IP, las que los navegadores usan para cargar páginas de internet [15]. La idea es permitir

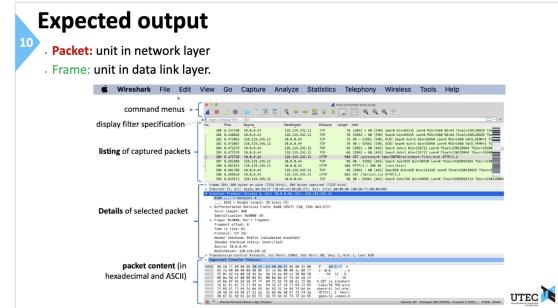


Figura 9: Vista de capturador de paquetes en Wireshark. Se tiene el command menus, para accesos, el display filter specification para filtrar protocolos (por ejemplo tcp), el listing of captured packets (en verde) para ver los paquetes reconocidos. Al presionar alguno vemos su detalle en Details of selected packet, y mas abajo podemos ver su contenido en hexadecimal (packet content). [14]

que las personas puedan ingresar palabras normales a sus navegadores, sin necesidad de conocer las direcciones IP para acceder a los diferentes sitios web. Por ejemplo, para que una persona simplemente pueda buscar [www.google.com](http://www.google.com) y no por su IP 142.251.0.139.

También, puede parecer relevante la idea de cómo se crean o registran dominios en Perú, específicamente para .pe, .edu y .com. Note que, dado el país, lo enfocaremos a .pe [16], .edu.pe [17] y .com.pe [16]. En sí, de manera generalizada, para cada uno de estos dominios el proceso abarca desde una fase de selección, hasta una renovación cada cierto tiempo tras el registro. Se puede ver dicho proceso para .pe, .edu.pe y .com.pe en la Figura 10, Figura 11 y Figura 12 respectivamente.

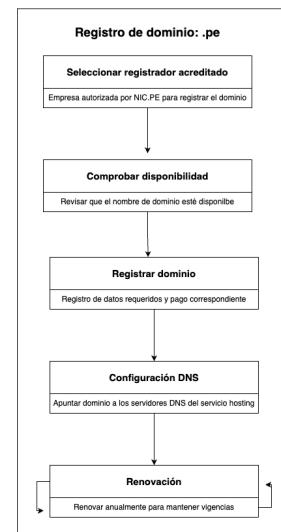


Figura 10: Flujo de registro de dominio para .pe

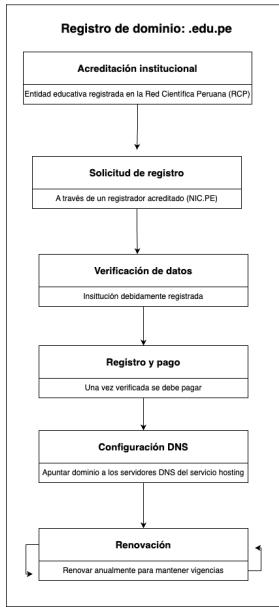


Figura 11: Flujo de registro de dominio para .edu.pe

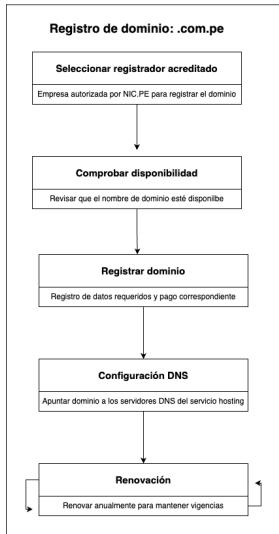


Figura 12: Flujo de registro de dominio para .com.pe

### I-B. Estado del Arte

Revisar la bibliografía involucrada nos puede dar una visión y noción mucho más clara de la importancia de DNS y también del software Wireshark. Haciendo una revisión detallada, vemos que una fracción mayoritaria de los trabajos hacen hincapié en temas de seguridad involucrando este protocolo y software mencionados, enfocándose principalmente en detección de amenazas, actividad sospechosa, análisis de tráfico, y demás aspectos relacionados. El trabajo *Investigating DHCP and DNS Protocols Using Wireshark* [18] trata bastante estas nociones. Explora que aunque existen diversas formas de mitigar ataques en las capas de aplicación, transporte y red, hacer ello en la capa de enlace de datos es una tarea

compleja, ya que no cuenta con una seguridad adecuada. Dado que los protocolos de DHCP y DNS son los que más operan en la capa de enlace de datos, dicha investigación estudia estos protocolos, capturando sus paquetes con la ayuda de Wireshark. Se analiza como se asigna una IP desde un servidor DHCP y como se utiliza DNS para resolver una URL en una dirección IP, identificando que son susceptibles a ataques de spoofing y envenenamiento de caché, y proponiendo esquemas de autenticación para mitigarlos.

Luego, el estudio *DNS over HTTPS Traffic Analysis and Detection* [19] también retoca estas ideas de seguridad, pero desde un punto distinto; en sí la desventaja de emplear un cifrado específico. Explica que DNS sobre HTTPS (DoH) básicamente proporciona un mapeo de solicitudes y respuestas DNS tradicionales dentro de mensajes HTTP encapsulados con TLS. Por tanto, DoH muestra tanto los beneficios criptográficos de TLS y de camuflar las comunicaciones como tráfico web común, y por tanto existen bastantes desafíos para identificar este tipo de comunicaciones en redes. Aborda lo anterior empleando Wireshark y da una visión opuesta de lo que esperaríamos de algo positivo como cifrados y ciberseguridad.

Por último, *Monitoring Security of Enterprise Hosts via DNS Data Analysis* [20] vuelve a un enfoque más tradicional, y sobre la susceptibilidad de protocolos como DNS, y la necesidad de esquemas para evitarlo, y de monitoreo constante. En dicho estudio se hace captura y análisis de más de 10 mil millones de paquetes de tráfico DNS, y a través de estos se ilustran muchos ataques como filtración de datos y comunicación de comando y control (C&C) a través de DNS. En respuesta a esto, se propone entrenar modelos de aprendizaje automático para detectar ello, y lograr mitigarlo.

Este pequeño extracto de la bibliografía nos da esta noción del involucramiento del protocolo DNS en prácticas y detección de seguridad, apoyándose del software Wireshark, que se usará en esta experiencia.

## II. DESARROLLO Y RESULTADOS

El desarrollo de esta experiencia se dividió en 7 fases principales, las cuales iremos detallando una por una en este informe. Para cada sección mostraremos por ítems las preguntas del laboratorio y luego el detalle de la experiencia y las respuestas.

### II-A. Converting a URL to an IP address using DNS

Para esta sección, trabajamos principalmente con la terminal de nuestro sistema operativo; en este caso, fueron dos sistemas Linux y uno macOS. A su vez, realizamos

la experiencia usando 2 URLs: [www.icann.net](http://www.icann.net) y [www.unsplash.com/](http://www.unsplash.com/).

### 1. What is the IP address of [www.icann.net](http://www.icann.net)?

Para este primer ítem lo que hicimos fue escribir el siguiente comando en la terminal **ping www.icann.net** y obtuvimos el siguiente resultado (Figura 13). Aquí podemos ver que la dirección IP de dicha URL es **192.0.43.22**.

```
+ ~ ping www.icann.net
PING www.icann.net (192.0.43.22) 56(84) bytes of data.
64 bytes from 43-22.any.icann.org (192.0.43.22): icmp_seq=1 ttl=240 time=107 ms
64 bytes from 43-22.any.icann.org (192.0.43.22): icmp_seq=2 ttl=240 time=95.4 ms
64 bytes from 43-22.any.icann.org (192.0.43.22): icmp_seq=3 ttl=240 time=91.0 ms
64 bytes from 43-22.any.icann.org (192.0.43.22): icmp_seq=4 ttl=240 time=92.5 ms
64 bytes from 43-22.any.icann.org (192.0.43.22): icmp_seq=5 ttl=240 time=91.5 ms
```

Figura 13: Ping a [www.icann.net](http://www.icann.net)

### 2. When you ping this address, do you get the same IP address as in the example or a different IP address? Why?

Antes de poder responder a esta pregunta tenemos que tener en cuenta que previamente se nos pidió ejecutar el siguiente comando en la terminal **ping www.unsplash.com**. El resultado de esta operación lo podemos observar en la Figura 14. Aquí podemos ver que la dirección IP es **199.232.133.181** la cual es diferente a la primera dirección IP que obtuvimos en el caso anterior. La razón por la que estas son distintas puede deberse a que cada url pertenece a un dominio distinto, por lo que cada una devolverá un ping distinto que es en dónde se encuentra alojado. Por otro lado, esto también se puede deber a que la url [www.unsplash.com](http://www.unsplash.com) usa el servicio Fastly. Este, lo que hace es redirigir la consulta al nodo más cerca al usuario que hace la petición, dicho nodo va a tener su propia dirección IP como dice en la documentación de Fastly [21].

```
+ ~ ping www.unsplash.com
PING p.shared.global.fastly.net (199.232.133.181) 56(84) bytes of data.
64 bytes from 199.232.133.181: icmp_seq=1 ttl=58 time=8.40 ms
64 bytes from 199.232.133.181: icmp_seq=2 ttl=58 time=7.58 ms
64 bytes from 199.232.133.181: icmp_seq=3 ttl=58 time=6.71 ms
^C
--- p.shared.global.fastly.net ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 6.710/7.563/8.397/0.688 ms
```

Figura 14: Ping a [www.unsplash.com](http://www.unsplash.com)

### 3. Type the IP address you got when you pinged [www.unsplash.com](http://www.unsplash.com) into a browser. Does the Web site show up? Why or why not? Investigate if necessary.

Al escribir la IP que obtuvimos (**199.232.133.181**) en el navegador no se renderizó ningún sitio web como se puede observar en la Figura 15. Esto se debe principalmente al uso del servicio Fastly. Como se menciona en su documentación, si un dominio no ha sido registrado en la plataforma de este servicio, al tratar de acceder mediante la IP generará el error “unknown domain” [22].



Figura 15: Respuesta del browser al ingresar la IP de [www.unsplash.com](http://www.unsplash.com) en el navegador.

### II-B. Observing DNS lookup using the nslookup command on a Web site

Para esta sección también usamos la terminal de nuestro sistema, pero en este caso usamos el comando **nslookup**. Luego usamos este comando junto con la siguiente dirección URL [www.unsplash.com](http://www.unsplash.com) de la siguiente manera **nslookup www.unsplash.com** y obtuvimos el resultado que se ve en la Figura 16

```
+ ~ nslookup www.unsplash.com
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
www.unsplash.com canonical name = p.shared.global.fastly.net.
Name: p.shared.global.fastly.net
Address: 151.101.193.181
Name: p.shared.global.fastly.net
Address: 151.101.65.181
Name: p.shared.global.fastly.net
Address: 151.101.129.181
Name: p.shared.global.fastly.net
Address: 151.101.1.181
```

Figura 16: Respuesta de la terminal al comando ‘nslookup www.unsplash.com’

### 1. What is the DNS that you are using? Is it a Local? root? TLD? or authoritative?

En mi caso estoy usando un DNS local, esto lo puedo observar en la Figura ?? ya que se puede ver que la dirección del servidor de consulta es del localhost **127.0.0.53**.

### 2. What is the IP address(es) translated when you search [www.unsplash.com](http://www.unsplash.com)

Este ítem también lo podemos responder basándonos en la Figura 16, ahí podemos observar que las direcciones IP que traducen a los dominios son las siguientes:

- 151.101.193.181
- 151.101.65.181
- 151.101.129.181
- 151.101.1.181

### 3. Is it the same IP address that appears with the ping command?

Sí y esto lo podemos observar en la Figura 17 y la Figura 18. Como se puede ver hicimos múltiples ejecuciones del comando **ping www.unsplash.com**. Esto lo hicimos para verificar qué direcciones IP aparecían en cada llamado del ping porque queríamos comprobar que aparecieran todas las direcciones que nos mostraba el comando **nslookup** como se ve en la Figura 16. Al ver que sí se usaban todas las direcciones IP de manera aleatoria para responder al llamado del comando ping, investigamos el por qué

había múltiples IP de respuesta. Vimos que al ser un servicio CDN, en este caso *fastly*, este distribuye el contenido en distintos servidores para dividir la carga.

```
~ ping www.unsplash.com
PING p.shared.global.fastly.net (151.101.65.181) 56(84) bytes of data.
64 bytes from 151.101.65.181: icmp_seq=1 ttl=56 time=5.52 ms
64 bytes from 151.101.65.181: icmp_seq=2 ttl=56 time=19.7 ms
64 bytes from 151.101.65.181: icmp_seq=3 ttl=56 time=6.27 ms
...
-- p.shared.global.fastly.net ping statistics --
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 5.521/10.483/19.666/6.496 ms

~ ping www.unsplash.com
PING p.shared.global.fastly.net (151.101.1.181) 56(84) bytes of data.
64 bytes from 151.101.1.181: icmp_seq=1 ttl=56 time=5.40 ms
64 bytes from 151.101.1.181: icmp_seq=2 ttl=56 time=6.82 ms
64 bytes from 151.101.1.181: icmp_seq=3 ttl=56 time=5.96 ms
64 bytes from 151.101.1.181: icmp_seq=4 ttl=56 time=6.59 ms
...
-- p.shared.global.fastly.net ping statistics --
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 5.396/6.193/6.824/0.558 ms

~ ping www.unsplash.com
PING p.shared.global.fastly.net (151.101.129.181) 56(84) bytes of data.
64 bytes from 151.101.129.181: icmp_seq=1 ttl=56 time=8.42 ms
64 bytes from 151.101.129.181: icmp_seq=2 ttl=56 time=5.46 ms
...
-- p.shared.global.fastly.net ping statistics --
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 5.455/6.935/8.415/1.480 ms
```

Figura 17: Ejecución múltiple del comando ‘ping www.unsplash.com’

```
* ~ ping www.unsplash.com
PING p.shared.global.fastly.net (151.101.193.181) 56(84) bytes of data.
 64 bytes from 151.101.193.181: icmp_seq=1 ttl=56 time=5.59 ms
```
-- p.shared.global.fastly.net ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.592/5.592/5.592/0.000 ms
```

Figura 18: Ejecución del comando ‘ping www.unsplash.com’.

4. *In the command prompt, type the IP address of the Unsplash Web server you just found. What result do you get?*

Para esta parte ingresamos primero el comando **nslookup** sin ningún argumento extra. Luego, en la línea donde aparece ‘>’ tipeamos la primera IP de Unsplash (**151.101.65.181**) y vimos que la respuesta era un *server can't find 181.65.101.151.in-addr.arpa: NXDOMAIN* como se ve en la Figura 19.

```
→ ~ nslookup  
> 151.101.65.181  
** server can't find 181.65.101.151.in-addr.arpa: NXDOMAIN  
>
```

Figura 19: Respuesta de tipar 151.101.65.181 en el *comand prompt*.

5. Now perform the same experience, but using the cisco.com site. Using the nslookup command, insert the IP address of the page. Does it differ from the previous result? Why? (Investigate if necessary to answer in the report).

Para este ítem realizamos el mismo proceso. Primero ejecutamos el comando **nslookup www.unsplash.com** y el resultado obtenido se puede ver en la Figura 20. Luego, agarramos la IP **72.163.4.185** y la tipeamos luego de ejecutar **nslookup** sin ningún argumento y obtuvimos lo que se ve en la Figura 21. Vemos que la respuesta es distinta en comparación al ítem previo, esto se debe a que no se ha asociado un DNS PTR record. El DNS

PTR record, como se menciona en Cloudflare [23], es un puntero que asocia un nombre de dominio a una dirección IP, esto es lo que permite que al hacer un nslookup y luego la dirección IP, cisco.com si nos devuelva el nombre de dominio ya que esta página si tiene asignado un DNS PTR para su IP. En cambio, como se puede ver www.unsplash.com no tiene asignado dicho DNS PTR por lo cual no encuentra un nombre de dominio asociado a dicha IP.

```
→ ~ nslookup cisco.com
Server:          127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:  cisco.com
Address: 72.163.4.185
Name:  cisco.com
Address: 2001:420:1101:1::185
```

Figura 20: Respuesta de la terminal al comando ‘nslookup cisco.com’.

```
[+ ~ nslookup  
> 72.163.4.185  
185.4.163.72.in-addr.arpa      name = redirect-ns.cisco.com.  
  
Authoritative answers can be found from:
```

Figura 21: Respuesta de tipar 72.163.4.185 en el *comand prompt*.

6. Now perform the same experiment, but using the [www.cisco.com](http://www.cisco.com) page. Does the resulting IP address change? Why? (Investigate if necessary to answer in the report).

Como se puede observar en la Figura 22 la respuesta en este caso es distinta a la del ítem anterior donde usábamos **cisco.com** nada más, a su vez la dirección IP también es distinta. Según muestra Cisco en su guía de resolución de dominios servidos por *akamai*, cuando un sitio web apunta a subdominios con CNAME's como *edgesuite.net*, *akamai.net*, *edgekey.net*, *amakaiedge.net*, entre otros, como se ve en la Figura 22, es porque está usando un servicio CDN *akamai*. Este servicio permite que los usuarios puedan resolver sus peticiones accediendo a los nodos CDN más cercanos a su ubicación geográfica, por lo que las direcciones IP van a variar dependiendo de esto [24].

```
~ nslookup www.cisco.com
Server: 192.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
www.cisco.com canonical name = www.cisco.com.akadns.net.
www.cisco.com.akadns.net canonical name = wwwds.cisco.com.edgekey.net.
wwwds.cisco.com.edgekey.net canonical name = wwwds.cisco.com.edgekey.net.globalredir.akadns.net.
Name: e2087.dsca.akamaiedge.net
Address: 23.0.234.108
Name: e2087.dsca.akamaiedge.net
Address: 2000:208:2094:181::b33
Name: e2087.dsca.akamaiedge.net
Address: 2000:208:2094:181::b33
```

Figura 22: Respuesta de la terminal al comando ‘nslookup www.cisco.com’.

7. You can also use it to translate IP addresses to domain names. Using the nslookup tool, write down the IP addresses associated with www.google.com.
- Para este ítem lo que hicimos fue ejecutar en la terminal el comando **nslookup www.google.com** y obtuvimos las diversas direcciones IP asociadas a los dominios de google como se puede ver en la Figura 23. Las direcciones IP son las siguientes:

- 142.251.0.147
- 2800:3f0:4003:c01::63
- 2800:3f0:4003:c01::69
- 2800:3f0:4003:c01::68
- 2800:3f0:4003:c01::67

```
→ ~ nslookup www.google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.251.0.147
Name:   www.google.com
Address: 2800:3f0:4003:c01::63
Name:   www.google.com
Address: 2800:3f0:4003:c01::69
Name:   www.google.com
Address: 2800:3f0:4003:c01::68
Name:   www.google.com
Address: 2800:3f0:4003:c01::67
```

Figura 23: Respuesta de la terminal al comando ‘nslookup www.google.com’.

8. Below the addresses, in addition to the IP address, you will see numbers separated by double dots. What kind of address is this? (Investigate if necessary).

Estas direcciones IP, según la documentación de IBM [25], son pertenecientes a las direcciones en formato IPv6, donde los (:) son una abreviación a una serie de 0's, es decir que en lugar de escribir, por ejemplo 3:0000:63, la abreviación sería 3::63

#### II-C. Using ipconfig

Para esta sección usamos la terminal de nuestro sistema, usando los siguientes comandos **ifconfig -a** y **ip route** con la finalidad de poder identificar las interfaces de red disponibles, así como las rutas y la puerta de enlace predeterminada configuradas en nuestros sistemas.

1. What is the current IP address, subnet mask, and default gateway of your host? (Use ipconfig /all and interpret the output.)

Para la respuesta de este ítem se usó el comando **ifconfig -a** por el sistema operativo de nuestras máquinas. Este comando nos proporcionaba todas las interfaces de red disponibles de nuestras máquinas,

por lo que fue necesario identificar cuál era el controlador que estábamos usando. Para este caso, la interfaz en uso ‘wlp0s20f3’ de la Figura [24], que corresponde al controlador de red inalámbrica (Wi-Fi).

```
wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.1.164 netmask 255.255.255.0 broadcast 10.0.1.255
inet6 fe80::991e:4622:a44:a95d prefixlen 64 scopeid 0x20<link>
ether f4:4a:75:49:48:5f txqueuelen 1000 (Ethernet)
RX packets 636952 bytes 788907169 (752.3 MiB)
RX errors 0 dropped 985 overruns 0 frame 0
TX packets 175496 bytes 87470657 (83.4 MiB)
TX errors 0 dropped 7 overruns 0 carrier 0 collisions 0
```

Figura 24: Información del ‘ifconfig -a’

En base a esta información de la Figura [24], podemos obtener que la IP address es **10.0.1.164** y la subnet mask es **255.255.255.0**. Realizando un análisis a profundidad, se descubrió que, en base a la dirección IP y la máscara de subred, la puerta de enlace predeterminada sería **10.0.1.1**. No obstante, **ifconfig** muestra la configuración local de la interfaz de red y no incluye directamente la puerta de enlace, la cual puede variar según la configuración de la red. Una forma de validar la puerta de enlace predeterminada es con el comando **ip route**, ya que este almacena explícitamente dicha configuración en nuestro sistema. Así que, en base a esa información de la Figura [25] el puerto de enlace predeterminado es **10.0.1.1**.

```
luisd ➔ ip route | grep default
default via 10.0.1.1 dev wlp0s20f3 proto dhcp src 10.0.1.164 metric 600
```

Figura 25: Información del ‘ip route’

2. List the DNS servers configured on your host. Are they static or assigned via DHCP? How can you tell?

Para poder obtener los servidores DNS configurados en el sistema, había que revisar un archivo llamado ‘resolv.conf’, el cual se encontraba guardado en un directorio de configuración del sistema como se puede ver en la Figura [26].

```
cat /etc/resolv.conf
```

Figura 26: Directorio de configuración

Al visualizar el contenido del archivo, se podía observar cierta información de servidores DNS Figura [27], la cual no se podía determinar con exactitud si era estática o generada por vía DHCP. Por lo tanto, la forma de comprobarlo era observar cómo cambiaba esa lista al cambiar de red Wi-Fi. Al realizar el experimento de cambiar de red, la información del archivo cambió completamente Figura [28], lo que nos llevó a la conclusión de que estos servidores DNS eran asignados por alguna vía DHCP.

```
luisd cat /etc/resolv.conf
# Generated by NetworkManager
search mecsa.com.pe
nameserver 10.0.1.5
```

Figura 27: Servidores DNS de la red 1

```
luisd cat /etc/resolv.conf
nameserver 192.168.125.233
nameserver 2800:4b0:4305:5cbe::16
```

Figura 28: Servidores DNS de la red 2

```
br-f7b05ca4ad4: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.19.0.1 netmask 255.255.0.0 broadcast 172.19.255.255
        ether e2:ab:7f:03:16:94 txqueuelen 0 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 1624 overruns 0 carrier 0 collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.255.0 broadcast 172.17.255.255
        ether f6:59:89:04:56:68 txqueuelen 64 scopeid 0x20<link>
        ether T6:59:89:04:56:68 txqueuelen 0 (Ethernet)
        RX packets 8 bytes 317 (317.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2845 bytes 384585 (375.5 KIB)
        TX errors 0 dropped 1 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        ether ::1 txqueuelen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
        RX packets 472 bytes 35832 (34.9 KIB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 472 bytes 35832 (34.9 KIB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vethbdb8473: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10:0:1:164 netmask 255.255.255.0 broadcast 10:0:1.255
        ether 9a:bd:97:65:83:c3 txqueuelen 0 (Ethernet)
        RX packets 8 bytes 429 (429.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 3640 bytes 468793 (457.8 KIB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.1.164 netmask 255.255.255.0 broadcast 10.0.1.255
        ether 9a:bd:97:65:83:c3 txqueuelen 1000 (Ethernet)
        RX packets 636952 bytes 788907169 (752.3 MiB)
        RX errors 0 dropped 985 overruns 0 frame 0
        TX packets 175496 bytes 87478657 (83.4 MiB)
        TX errors 0 dropped 7 overruns 0 carrier 0 collisions 0
```

Figura 29: Información del ‘ifconfig -a’

### 3. What type of network adapter is used to connect to the internet? Wired, wireless, or virtual?

Para responder este ítem se uso el comando **ifconfig -a**, el cual mostraba todas las interfaces de red disponibles Figura [29]. Aunque el sistema estaba conectado a través de una red Wi-Fi, era necesario confirmar cuál de las interfaces correspondía a dicha conexión. Al realizar un análisis del resultado del comando, se podían apreciar una gran variedad de interfaces, incluyendo algunas generadas por Docker (como docker0) y otras asociadas a bridges virtuales (br...), las cuales podían descartarse por su nombre y propósito. Debido a que no correspondían a interfaces físicas de conexión a Internet. La forma más precisa de confirmar cual era la interfaz de red Wi-Fi era en base a dos criterios: la convención de nombres y la actividad de red. La interfaz ‘wlp0s20f3’ seguía la nomenclatura típica de interfaces inalámbricas (wi...) y además presentaba un tráfico significativo de paquetes transmitidos y recibidos Figura [29], lo que indicaba que era la única en uso real. Por ello, se concluyó que ‘wlp0s20f3’ era la interfaz correspondiente a la conexión Wi-Fi activa del sistema .

### 4. Before flushing the DNS cache, what DNS entries are currently stored on your machine? Use ipconfig /displaydns and summarize the results.)

Para realizar esta experiencia se utilizo el comando **sudo resolvectl show-cache** el cual mandaba muchas entradas DNS guardadas en la maquina Figura [30]. En él se observaban dominios de conectividad con Ubuntu, servicios de GitHub Copilot y telemetría, así como dominios auxiliares y de verificación de conectividad. No obstante, debido a que su sistema no contaba con un servidor DNS configurado localmente Figura [31]. Esto significa que no se utilizaba un servicio de caché local para resolver nombres de dominio. Lo que provocaba que cada vez que el sistema necesitaba acceder a un dominio (por ejemplo, www.google.com), debía realizar la consulta directamente a un servidor DNS externo, sin guardar resultados en una caché local. Provocando que las consultas DNS se repitan incluso para dominios ya visitados, lo que puede hacer las conexiones más lentas.

```

desarrollador at servidordeav2 in [/home/desarrollador]
[5:40:47] > sudo resolvectl show-cache

Scope protocols dns ifindex=2 ifname=enps192
gallerycdn.vsassets.io.edgesuite.net IN CNAME a125.dscr.akamai.net
ubuntu.23sl.ufpr.br IN A 200.236.31.4
connectivity-check.ubuntu.com IN A 185.125.190.98
connectivity-check.ubuntu.com IN A 185.125.190.96
connectivity-check.ubuntu.com IN A 185.125.190.17
connectivity-check.ubuntu.com IN A 91.189.91.96
connectivity-check.ubuntu.com IN A 91.189.91.48
connectivity-check.ubuntu.com IN A 91.189.91.49
connectivity-check.ubuntu.com IN AAAA 2001:67c:1562::23
connectivity-check.ubuntu.com IN AAAA 2001:67c:1562::22
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4000::1::97
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4000::1::2b
connectivity-check.ubuntu.com IN AAAA 2001:67c:1562::24
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4000::1::2a
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4000::1::23
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4000::1::96
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4002::1::196
connectivity-check.ubuntu.com IN AAAA 2620:2d4:4000::1::98
api.individualithubcopilot.com IN CNAME api.githubcopilot.com
api.githubcopilot.com IN QNAME g1b-db52c2cf8be544.github.com
copilot-telemetry-service.githubusercontent.com IN QNAME g1b-db52c2cf8be544.github.com
api-gateway.p.tabnine.com IN A 34.123.33.186

Scope protocol=dns

```

Figura 30: Información del ‘sudo resolvectl show-cache’

```

==== AUTHENTICATING FOR org.freedesktop.resolve1.dump-statistics ====
Authentication is required to dump statistics.
Authenticating as: Loza05
Password:
==== AUTHENTICATION COMPLETE ====
Transactions          Current Transactions:  0
   Total Transactions: 1359
Cache                Current Cache Size:  60
   Cache Hits: 750
   Cache Misses: 594
Failure Transactions   Total Timeouts:  9
   Total Timeouts (Stale Data Served): 0
   Total Failure Responses: 0
   Total Failure Responses (Stale Data Served): 0
DNSSEC Verdicts        Secure:  0
   Insecure: 0
   Bogus: 0
   Indeterminate: 0
+ ~ resolvectl flush-caches
+ ~ resolvectl statistics

==== AUTHENTICATING FOR org.freedesktop.resolve1.dump-statistics ====
Authentication is required to dump statistics.
Authenticating as: Loza05
Password:
==== AUTHENTICATION COMPLETE ====
Transactions          Current Transactions:  0
   Total Transactions: 1377
Cache                Current Cache Size:  18
   Cache Hits: 750
   Cache Misses: 612
Failure Transactions   Total Timeouts:  9
   Total Timeouts (Stale Data Served): 0
   Total Failure Responses: 0
   Total Failure Responses (Stale Data Served): 0
DNSSEC Verdicts        Secure:  0
   Insecure: 0
   Bogus: 0
   Indeterminate: 0

```

Figura 32: Información del ‘resolvectl flush-caches’

```

luisd  ps aux | grep -E 'system-resolve|dnsmasq|nsqd|unbound'
root    422820  0.0  0.0    0    0 ? 10:05 0:00 [kworker/u32:2-events_unbound]
root    654791  0.0  0.0    0    0 ? 10:11 0:00 [kworker/u32:3-events_unbound]
luisd   856283  0.0  0.0  6616 4096 pts/2  S+ 10:17 0:00 grep -E system-resolve|dnsmasq|nsqd|unbound

```

Figura 31: Ausencia de servidor DNS local

5. **What happens when you run ipconfig /flushdns? How does it affect the DNS cache content?**
- Para este item se uso el comando **resolvectl flush-caches**. Al ejecutar este comando, se realizaba una limpieza de las entradas DNS almacenadas en el sistema. Como se puede ver en la Figura [32], ocurrió una reducción en el tamaño de la caché, lo cual significa que todos los registros DNS almacenados habían sido borrados. Esto provocaba que, al realizar una petición a un dominio, el sistema tuviera que consultar al servidor DNS externo para obtener la dirección IP correspondiente; lo que podía generar ligeras demoras al acceder por primera vez a sitios web que ya habían sido visitados anteriormente.

#### *II-D. DNS lookup using the nslookup command on mail servers*

Para este item realizamos se ejecuto primero el comando **nslookup**, una vez ejecutado se tenia q setear **set type=mx** con el fin de poder identificar los servidores de correo. Para finalmente ejecutar **www.cisco.com | utec.edu.pe | www.google.com** Figura [33]

```

luisd  nslookup
> set type=mx
> www.cisco.com | utec.edu.pe | www.google.com
Server:  200.46.225.130
Address: 200.46.225.130#53
Non-authoritative answer:
www.cisco.com canonical name = www.cisco.com.akadns.net.
www.cisco.com.akadns.net canonical name = wwws.cisco.com.edgekey.net.
wwws.cisco.com.edgekey.net canonical name = wwws.cisco.com.edgekey.net.globalredir.akadns.net.
wwws.cisco.com.edgekey.net.globalredir.akadns.net canonical name = e2067.dsc.akamaiedge.net.
Authoritative answers can be found from:
dsca.akamaiedge.net
origin = dsca.akamaiedge.net
mail = hostmaster.akamai.com
serial = 1745561712
refresh = 10000
retry = 1000
expire = 1000
minimum = 1000
> |

```

Figura 33: Información del ‘nslookup con set type=mx’

#### *1. Which server will be contacted first when sending email to cisco.com?*

El servidor que será contactado primero al enviar un correo a ‘cisco.com’ es ‘alln-mx-01.cisco.com’. Esto debido a que este servidor posee el valor de **MX** más bajo (10) en comparación con los otros Figura [34]. En los registros de MX, cuanto menor es el valor, mayor es la prioridad.

```
Non-authoritative answer:
cisco.com      mail exchanger = 30 aer-mx-01.cisco.com.
cisco.com      mail exchanger = 10 alln-mx-01.cisco.com.
cisco.com      mail exchanger = 20 rcdn-mx-01.cisco.com.
```

Figura 34: Listado con los valores de prioridad de los registros MX

## 2. Perform the same experience for gmail.com which server will be contacted first when email is sent to gmail.com?

Al realizar la consulta DNS con el comando **nslookup** para el dominio ‘gmail.com’, se obtuvo una lista de registros MX asociados, con sus valores de prioridad. Según la Figura [35], el servidor con mayor prioridad es ‘gmail-smtp-in.l.google.com’ con un valor de **5**. Lo que significa que este sera el primer servidor contactado al enviar un correo a ‘gmail.com’.

```
> gmail.com
Server:      10.0.1.5
Address:     10.0.1.5#53

Non-authoritative answer:
gmail.com      mail exchanger = 5 gmail-smtp-in.l.google.com.
gmail.com      mail exchanger = 10 alt1.gmail-smtp-in.l.google.com.
gmail.com      mail exchanger = 20 alt2.gmail-smtp-in.l.google.com.
gmail.com      mail exchanger = 40 alt4.gmail-smtp-in.l.google.com.
gmail.com      mail exchanger = 30 alt3.gmail-smtp-in.l.google.com.

Authoritative answers can be found from:
gmail-smtp-in.l.google.com      internet address = 142.251.0.27
>_
```

Figura 35: Listado con los valores de prioridad de los registros MX

## 3. At the nslookup prompt, type exit to return to the normal PC command prompt. At the PC's command prompt, type ipconfig /all. Write down the IP addresses of all DNS servers used by the university.

Para este item se utilizo la inspección del archivo **/etc/resolv.conf** conectado a la red de la universidad. Al realizar la inspección, los servidores DNS utilizados por la universidad fueron Figura [36]

```
luisd ➔ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.100.2.21
nameserver 10.100.2.22
```

Figura 36: Servidores DNS utilizados por la universidad.

## II-E. DNS Tracing with Wireshark I - Browsing

Para esta sección primero empezamos limpiando nuestra caché con el siguiente comando en el caso de los usuarios de linux **resolvectl flush-caches**, en el caso del usuario de macOS, usaron los comandos **sudo dscacheutil -flushcache; sudo killall -HUP mDNSResponder**. Luego, limpiamos el caché de nuestro browser y entramos en Wireshark y pusimos el filtro dns (Figura37) y empezamos a capturar paquetes. Para esta experiencia nosotros usamos el link [www.ietf.org](http://www.ietf.org) y al hacer la captura de paquetes obtuvimos la información que se muestra en la Figura

38. Sin embargo, también probamos con [www.cisco.com](http://www.cisco.com) y vimos que la cantidad de paquetes era sumamente mayor, como se ve en la Figura 39. Esto tiene mucho que ver con lo analizado en la sección II-A de este informe, ahí vimos que dicho dominio trabaja con un sistema CDN que redirecciona las consultas, esto implica nuevas consultas DNS. A su vez, también se puede observar que hay consultas tipo A y tipo AAAA, lo que hace referencia a consultar direcciones IPv4 e IPv6 y esto también agrega más consultas DNS.

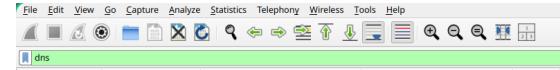


Figura 37: Sección de Wireshark donde se escribe el filtro a aplicar.

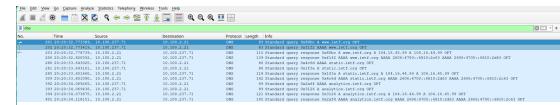


Figura 38: Sección de packet listing de wireshark con consulta www.ietf.org.

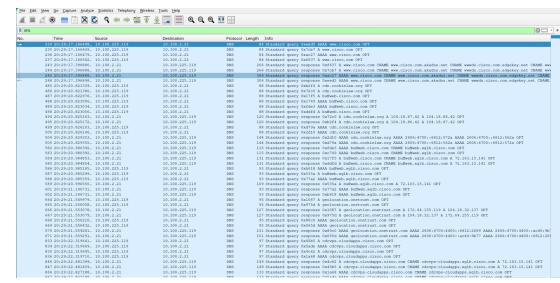


Figura 39: Sección de packet listing de wireshark con consulta www.cisco.com.

## 1. Analyze and explain how a DNS request is structured using Wireshark.

Para responder a este item accedimos a la sección *Domain Name System* del primer *query* que nos apareció en el filtro (Figura 40). Aquí podemos ver la estructura la cual contiene los siguientes datos: un ID que es el identificador único de dicha consulta, unos *FLAGS* los cuales nos dan información de la operación hecha, por ejemplo que es un *query* no una respuesta, que es una consulta estándar (*Opcode*), que se necesita resolver esta consulta de manera recursiva (*Recursion Desired*), entre otros; una sección *Questions* que nos indica cuántas consultas se han hecho, *Answer RRS* que indica si es que hay alguna respuesta, al ser esta una consulta no hay respuesta por lo que su valor es 0; una sección *Queries* que contiene la información de las consultas que se han hecho, por ejemplo el dominio, el tipo de consulta (en este caso A). Esta es la estructura principal de la consulta DNS.

```

▼ Domain Name System (query)
  Transaction ID: 0x60bc
  ▼ Flags: 0x0100 Standard query
    0..... .... = Response: Message is a query
    .000 0.... .... = Opcode: Standard query (0)
    .... 0.... .... = Truncated: Message is not truncated
    .... 0.... .... = Recursion desired: Do query recursively
    .... 0.... .... = Z: reserved (0)
    .... 0.... .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 1
  ▼ Queries
    - www.ietf.org: type A, class IN
      Name: www.ietf.org
      [Name Length: 12]
      [Label Count: 3]
      Type: A (1) (Host Address)
      Class: IN (0x0001)
  ▼ Additional records
    - <Root>: type OPT
      Name: <Root>
      Type: OPT (41)
      UDP payload size: 1472
      Higher bits in extended RCODE: 0x00
      EDNS0 version: 0
    > Z: 0x0000
      Data length: 0
  [Responses In: 203]

```

Figura 40: Estructura de una consulta DNS en Wireshark.

2. **To which IP address is the DNS query message sent? Examine the DNS query message. How many "questions" does this DNS message contain? How many ".answers" does it contain?**

Como podemos ver en la sección de *destination* en la Figura 41, la consulta DNS es enviada hacia la dirección **10.100.2.21**. A su vez, como podemos ver en la Figura 40 se solo una pregunta (*Question*) y contiene 0 respuestas (*Answer RRs*) ya que esta es sólo una consulta.

| No. | Time                   | Source      | Destination | Protocol | Length | Info                                     |
|-----|------------------------|-------------|-------------|----------|--------|------------------------------------------|
| 1   | 2012-08-16 09:10:23.71 | 10.100.2.21 | 10.100.2.21 | DNS      | 43     | Standard query 0x07d0 & www.ietf.org OPT |

Figura 41: Primera consulta DNS en Wireshark.

## II-F. Tracing DNS with Wireshark II - Nslookup

Para esta sección se hizo uso del comando **nslookup** con **wireshark**.

- Primero se ejecutó **nslookup www.mit.edu** Figura [42]

```

nslookup www.mit.edu
Server: 200.48.225.130
Address: 200.48.225.130#53

Non-authoritative answer:
www.mit.edu canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dsrb.akamaiedge.net.
Name: e9566.dsrb.akamaiedge.net
Address: 96.6.197.28
Name: e9566.dsrb.akamaiedge.net
Address: 2600:1419:4400:183::255e
Name: e9566.dsrb.akamaiedge.net
Address: 2600:1419:4400:1a8::255e

```

Figura 42: Información del 'nslookup www.mit.edu'

1. **What is the destination port for the DNS query message? What is the source port of DNS response message?** En la Figura[43] se puede observar que la consulta DNS inicial muestra que el 'Source Port' es el 51481 por parte de nuestro sistema y el 'Destination Port'es el 53 por parte del dominio 'www.mit.edu'.

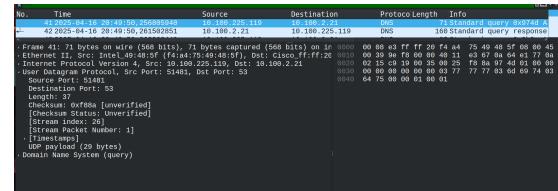


Figura 43: Wireshark capturando 'nslookup www.mit.edu'

2. **To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?**

En base a la Figura[43] se puede observar que la dirección IP que se envía el mensaje es la **10.100.2.21**. No obstante, necesitamos saber si es la dirección IP que corresponde a nuestro DNS local por defecto. Para averguar esto, debemos inspeccionar el archivo '/etc/resolv.conf'.

```

cat /etc/resolv.conf

# Generated by NetworkManager
nameserver 10.100.2.21
nameserver 10.100.2.22

```

Figura 44: Servidores DNS local usados por nuestro sistema

Segun la Figura [44] podemos afirmar que la dirección IP que se captura en la Figura [43] corresponde a nuestra IP de nuestro servidor DNS local por defecto.

3. **Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?**

El paquete Nº41 Figura [43] es una consulta DNS de tipo A hacia 'www.mit.edu'. Lo que significa que esta solicitando la dirección IPv4 asociada al nombre del dominio. Este mensaje no posee respuesta Figura [45].

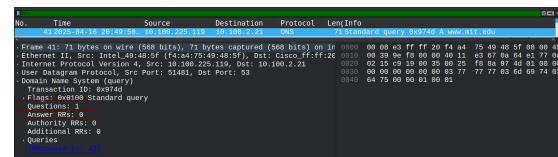


Figura 45: Paquete Nº41 de wireshark

4. **Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?**

El paquete Nº42 Figura [43] posee 3 respuestas Figura [46]. Las cuales indican que el dominio 'www.mit.edu' es un alias definido por un registro txtbfCNAME que apunta a 'www.mit.edu.edgekey.net'. A su vez, este dominio también es un alias, redirigiendo hacia

‘e9566.dscb.akamaiedge.net’, otro registro CNAME. Finalmente, este último dominio se resuelve mediante un registro de tipo A a la dirección IP **96.6.197.28**.

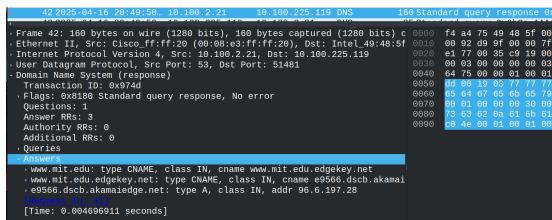


Figura 46: Paquete N°42 de wireshark

## 5. Provide a screenshot

Una screenshot de las consultas DNS son mostradas en la Figura [43]

Para los siguientes items se ejecuto **nslookup -type=NS mit.edu**

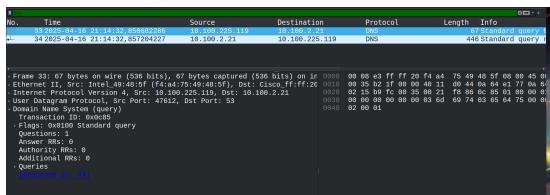


Figura 47: Wireshark capturando ‘nslookup –type=NS mit.edu’

## 6. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

En base a la Figura [47] se puede observar que la dirección IP que se envía el mensaje es la **10.100.2.21**. Y para validar que esta sea nuestra IP de nuestro servidor DNS local por defecto, nos basamos en la Figura [44].

## 7. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”? El paquete N°33 Figura [47] corresponde a una consulta DNS estándar (Standar query), indicado por la info y por flags que posee. Dentro de esta query se observa que hay una consulta y ninguna respuesta Figura [48]. Este tipo de consultas especificados con **NS**, indica que está solicitando la lista de servidores de nombres autorizados para el dominio ‘mit.edu’.

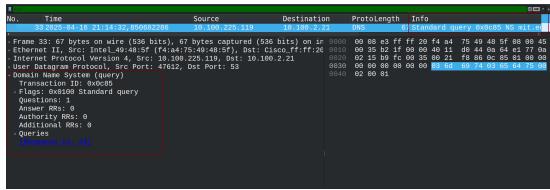


Figura 48: Paquete N°33 de wireshark

## 8. Examine the DNS response message. What MIT nameservers does the response message provide?

*Does this response message also provide the IP addresses of the MIT namesers?*

El paquete N°34 Figura [47] contiene un total de 8 registros NS para el dominio ‘mit.edu’. Estos registros indican los servidores de nombres autorizados para dicho dominio Figura [49]. Los servidores proporcionados son:

- asial.akam.net
- asia2.akam.net
- ns1-37.akam.net
- eur5.akam.net
- use2.akam.net
- use5.akam.net
- ns1-173.akam.net
- usw2.akam.net

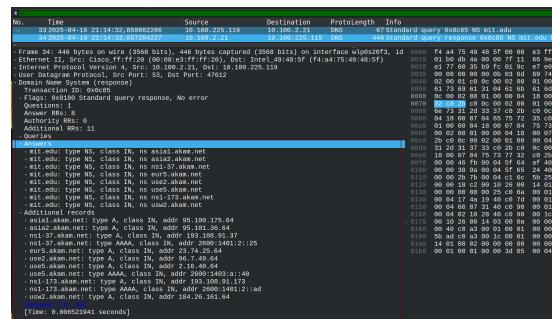


Figura 49: Paquete N°34 de wireshark

Además, el mensaje incluye una sección de registros adicionales, en la cual se proporcionan las direcciones IP (tipo A y AAAA) correspondientes a varios de estos servidores Figura [49]

## 9. Provide a screenshot

Una screenshot de las consultas DNS son mostradas en la Figura [47]

Para los siguientes items se ejecuto **nslookup www.aiit.or.kr bitsy.mit.edu**

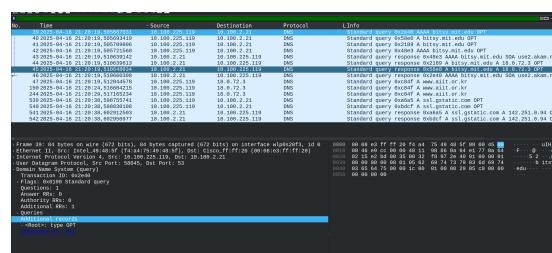


Figura 50: Wireshark capturando ‘nslookup www.aiit.or.kr bitsy.mit.edu’

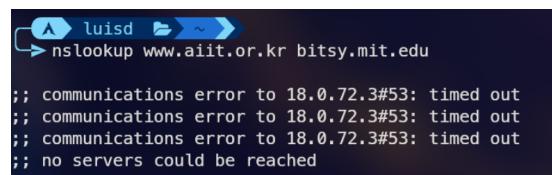


Figura 51: Ejecución nslookup www.aiit.or.kr bitsy.mit.edu

Durante la ejecución del comando, se produjo un error de comunicación con la dirección IP 18.0.72.3, puerto 53 Figura [51], generando múltiples mensajes de "timed out" finalizando con "no servers could be reached". Este comportamiento se origino debido a cómo funciona el comando **nslookup** cuando se le proporciona un segundo parámetro; este segundo argumento se interpreta como la dirección del servidor DNS al cual se desea realizar la consulta.

En este caso, se intentó contactar al dominio 'bitsy.mit.edu' como si fuera un servidor DNS. 'bitsy.mit.edu' es simplemente un hostname del MIT, pero no está configurado para aceptar ni responder consultas DNS en el puerto '53'. Por lo tanto, cuando **nslookup** intenta establecer una conexión con ese host para resolver el dominio 'www.aiit.or.kr', los intentos de comunicación fallan.

A diferencia de las otras consultas, en este caso se está forzando a que la consulta sea enviada a un servidor DNS específico ('bitsy.mit.edu'), el cual que no esté habilitado como servicio DNS.

**10. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?**

La consulta DNS fue enviada a la dirección IP **10.100.2.21**, como se puede ver en la Figura [50]. Y para validar que esta sea nuestra IP de nuestro servidor DNS local por defecto, nos basamos en la Figura [44].

**11. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?**

El paquete N° 39 corresponde a una solicitud de tipo 'AAAA', lo que indica que el cliente está solicitando la dirección IPv6 del dominio 'bitsy.mit.edu'. Contiene una pregunta y cero respuestas Figura [52]

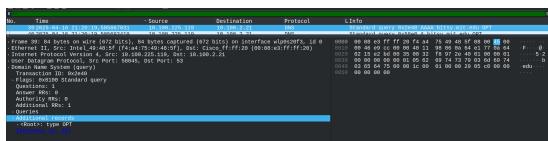


Figura 52: Paquete N°39 de wireshark

**12. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?**

El paquete N°46 no proporciona respuestas con servidores de nombres del MIT. La respuesta a la consulta 'bitsy.mit.edu' no contenía registros NS ni registros A o AAAA válidos. En su lugar, el servidor DNS respondió con un registro 'SOA' Figura [53] indicando que la zona está gestionada por 'use2.akam.net', sin incluir direcciones IP directas.

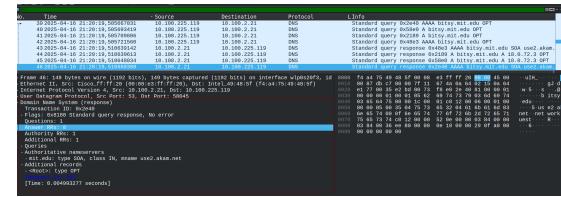


Figura 53: Paquete N°46 de wireshark

Esto reafirma de que 'bitsy.mit.edu' no esta diseñado para responder consultas DNS como servidor autoritativo, sino que es simplemente un nombre de dominio.

**13. Provide a screenshot**

Una screenshot de las consultas DNS son mostradas en la Figura [50] y sobre la ejecución del comando en la Figura [51]

**II-G. DNS Tracing with Wireshark - Browsing**

**1. What is the primary purpose of DNS?**

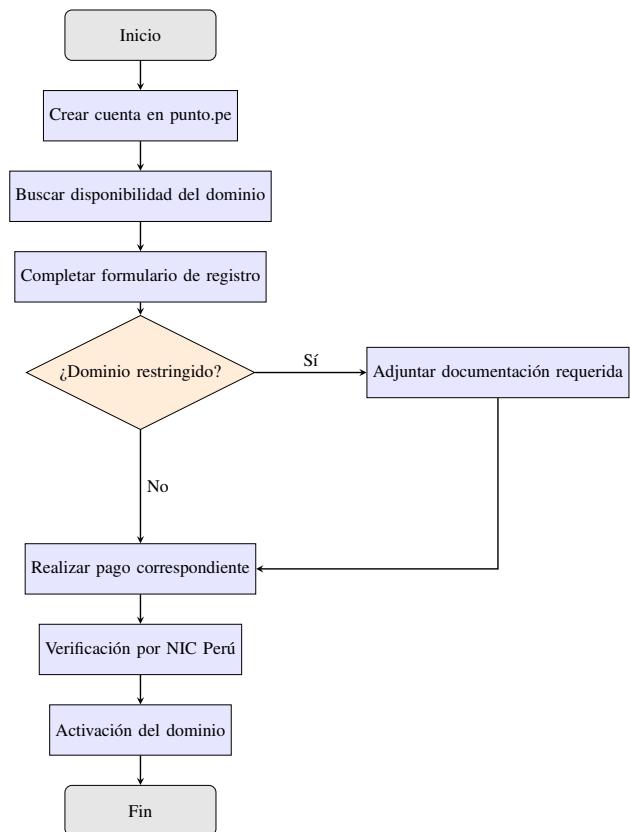
Como se menciona en la documentación técnica de Cloudflare [26] el principal propósito del DNS es traducir los nombres de dominio a direcciones IP. Esto es extremadamente útil ya que las computadoras se comunican mediante sus identificadores que son las IP, por lo tanto, si se quiere acceder a un recurso que está hosteado en una computadora X, se necesita de su IP. Como sería sumamente difícil para los humanos aprender una secuencia de números para cada recurso al que quieran acceder en el Internet, DNS se encarga de mapear estas IP a direcciones de dominio, las cuales son más entendibles para los humanos.

**2. Research how domain registration is carried out in Peru for the following domains: “.pe, “.edu, and “.com. Then explain the procedure using a block diagram.**

- El Dominio .pe es un dominio no restringido por lo que el proceso de registro es más sencillo que en el caso del dominio .edu que si es un dominio restringido. En ambos casos tienes que acceder al sitio web punto.pe, luego tienes que buscar la disponibilidad del dominio que deseas registrar, y completar un formulario con los datos que te piden. En el caso del dominio .edu, tienes que adjuntar cierta documentación que te piden para acreditar la institución a la cuálquieres darle ese dominio. Estos contienen información acerca del RUC de la institución, copia del documento de autorización otorgado por el Ministerio de Educación, entre otros. Seguido de eso, en ambos casos realizas el pago correspondiente y esperas a que la información sea verificada y se apruebe la solicitud que se ha hecho para obtener el dominio.

En el caso del dominio .com es muy similar el proceso sólo que esto lo puedes realizar

en páginas como **GoDaddy**, **DomWeb**, **PlanteaHosting** o **HostingPeru**.



### III. CONCLUSIONES

- En la sección II-A pudimos ver como el sistema DNS nos sirve para resolver una URL en una o más direcciones IP usando el comando **ping**. Se observó también como la infraestructura de los sitios web puede influir en las direcciones IP que devuelve, como en este caso lo fue Unsplash que al usar Fastly, un servicio CDN, distribuye las peticiones que se le hacen al servidor más cercano geográficamente para una mayor rapidez.
- En la sección II-B profundizamos acerca de como se resuelven los nombres de dominio a nivel del servidor DNS usando el comando **nslookup** en nuestras terminales. En la experiencia pudimos ver como el uso de un servicio CDN, como el que tiene Unsplash, asocia múltiples direcciones a un nombre de dominio, mostrando así su infraestructura distribuída. También, pudimos ver el uso del DNS inverso, como lo hace **cisco.com**, a través de lo que se conoce como PTR. Esto permitía a los sitios que tengan configurado esto resolver de direcciones IP a nombres de dominio.

- En la sección II-C se identificaron los parámetros de red del sistema, como la dirección IP, la máscara de subred y la puerta de enlace, utilizando los comandos **ifconfig -a** e **ip route**. Se determinó que la conexión activa era mediante una interfaz inalámbrica y que los servidores DNS eran asignados dinámicamente a través de DHCP, lo cual fue confirmado al observar cambios en el archivo de configuración ‘resolv.conf’ al cambiar de red. Luego se vio que la mayoría de los sistemas tienen un servicio de caché DNS local para mejorar las búsquedas; no obstante, puede haber sistemas donde dicho servicio haya sido desactivado. Finalmente, al ejecutar **resolvectl flush-caches** en un sistema con un servicio de caché DNS local, se evidenció la eliminación de las entradas DNS almacenadas, lo que obligará al sistema a realizar nuevas consultas para resolver dominios.
- En la sección II-D se identificaron los servicios de correo electrónico asociados a distintos dominios. Se observó que al enviar un correo a ‘cisco.com’ el primer servidor en ser contactado sería ‘alln-mx-01.cisco.com’ debido a su mayor prioridad, mientras que para ‘gmail.com’ el servidor de mayor prioridad fue ‘gmail-smtp-in.l.google.com’. Finalmente, se identificaron los servidores DNS utilizados por la universidad mediante la revisión del archivo ‘/etc/resolv.conf’, entendiendo de manera más precisa el funcionamiento de la infraestructura DNS y la selección prioritaria de servidores de correo.
- En la sección II-E nos centramos en el análisis de los paquetes DNS capturados por Wireshark, donde pudimos ver la estructura de una consulta DNS y como cada campo tenía un papel importante en toda la consulta. A su vez, en la comparación que hicimos entre los sitios www.ietf.org y www.cisco.com pudimos darnos cuenta de como un dominio que trabaja con el sistema CDN tiene que hacer una mayor cantidad de consultas DNS para poder resolverla finalmente en el servicio más cercano a dónde fue hecha la consulta, a su vez que también vimos que se hacen tanto consultas para IPv4(A) e IPv6(AAAA). Así pudimos ver que las servicios que usan estructuras distintas, como la CDN, afectan también en el flujo de las consultas DNS llegando a tener una cantidad mucho mayor en comparación a un servicio que no lo tiene.
- En la sección II-F se analizó el proceso de resolución DNS a nivel de red. Aquí se identificaron los puertos de origen y destino de las consultas DNS realizadas con el comando **nslookup**, también se identificó las direcciones IP utilizadas para poder verificar si correspondían al servidor DNS local del sistema. A su vez, se examinaron las consultas y respuestas DNS, donde se encontraron el uso de registros de tipo A, CNAME y NS, donde se podía entender el flujo de redirección de los dominios hasta su resolución final. También se

experimento que no todos los host aceptan consultas DNS, lo que profundizó la comprensión sobre el rol de los servidores autoritativos y la forma en que se gestionan las respuestas DNS ante consultas exitosas y fallidas.

## REFERENCIAS

- [1] Cloudflare, "¿Qué es el modelo OSI?", *Cloudflare*. [En línea]. Disponible en: <https://www.cloudflare.com/es-es/learning/ddos/glossary/open-systems-interconnection-model-osi/>
- [2] Proofpoint, "¿Qué es el modelo OSI? Definición, capas y más", *Proofpoint*. Disponible en: <https://www.proofpoint.com/es/threat-reference/osi-model>
- [3] Kaspersky, "¿Qué es una dirección IP y qué significa?", *Kaspersky Resource Center*. Disponible en: <https://www.kaspersky.es/resource-center/definitions/what-is-an-ip-address>
- [4] Webempresa, "¿Qué es un Dominio y cómo funciona?", *Webempresa*. Disponible en: <https://www.webempresa.com/hosting/que-es-dominio.html>
- [5] Cloudflare, "¿Qué es DNS? | Cómo funciona", *Cloudflare*. Disponible en: <https://www.cloudflare.com/es-es/learning/dns/what-is-dns/>
- [6] C. Williams, *DNS - Computer Networks*. Presentación basada en los libros de Kurose y de Peterson y Davie, UTEC - Universidad de Ingeniería y Tecnología, Lima, Perú.
- [7] Axarnet, "NSLOOKUP: Qué es y cómo usarlo en servidores DNS", *Axarnet*. Disponible en: <https://axarnet.es/blog/que-es-nslookup>
- [8] Microsoft, "ipconfig", *Microsoft Learn*. Disponible en: <https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/ipconfig>
- [9] Y. Fernández, "Máscara de subred: qué es y para qué sirve", *Xataka*, 22 de julio de 2022. Disponible en: <https://www.xataka.com/basics/mascara-subred-que-sirve>
- [10] ExpressVPN, "Cómo encontrar su dirección IP privada y la dirección de puerta de enlace predeterminada", *ExpressVPN*, 13 de septiembre de 2023. Disponible en: <https://www.expressvpn.com/es/support/troubleshooting/find-default-gateway/>
- [11] Juan Leonardo, "Comando ifconfig de Linux", *EXTASSIS NETwork*, 9 de enero de 2024. Disponible en: <https://extassisnetwork.com/tutoriales/comando-ifconfig-de-linux/>
- [12] Khan Academy, "Protocolo de control de transmisión (TCP)", *Khan Academy*. [En línea]. Disponible en: <https://es.khanacademy.org/computing/ap-computer-science-principles/the-internet/x2d2f703b37b450a3:transporting-packets/a/transmission-control-protocol--tcp> [Accedido: 15-abr-2025].
- [13] R. Altube, "Wireshark: Qué es y ejemplos de uso", *OpenWebinars*, 07-ene-2021. [En línea]. Disponible en: <https://openwebinars.net/blog/wireshark-que-es-y-ejemplos-de-uso/> [Accedido: 15-abr-2025].
- [14] Garias, "Wireshark Presentation", Lima, Perú: Universidad de Ingeniería y Tecnología (UTEC), Curso CS4054 – Redes y Comunicaciones, 2025.
- [15] FlashStart, "¿Por qué es importante el DNS?", *FlashStart*, 25 de junio de 2024. Disponible en: <https://flashstart.com/es/por-que-es-importante-el-dns/>
- [16] EuroDNS, "Comprobar disponibilidad de dominio", *EuroDNS*. Disponible en: <https://www.eurodns.com/es/buscar-dominios-libres>
- [17] Soy.pe, "Cómo registrar un dominio .edu.pe", *DominiosPeru.pe*, 5 de septiembre de 2020. Disponible en: <https://dominiosperu.pe/como-registrar-un-dominio-edu-pe/>
- [18] S. Naaz y F. A. Badroo, "Investigating DHCP and DNS Protocols Using Wireshark", *IOSR Journal of Computer Engineering*, vol. 18, no. 3, pp. 1–8, junio 2016. Disponible en: [https://www.researchgate.net/publication/316877345\\_Investigating\\_DHCP\\_and\\_DNS\\_Proocols\\_Using\\_Wireshark](https://www.researchgate.net/publication/316877345_Investigating_DHCP_and_DNS_Proocols_Using_Wireshark)
- [19] C. López Romera, "Análisis y detección del tráfico DNS sobre HTTPS", *Universitat Oberta de Catalunya*, 2020. Disponible en: <https://openaccess.uoc.edu/handle/10609/119946>
- [20] J. Ahmed, "Monitoring Security of Enterprise Hosts via DNS Data Analysis", *arXiv*, 18 de mayo de 2022. Disponible en: <https://arxiv.org/abs/2205.08968>
- [21] Fastly. (s.f.). *Introduction to CDN: What is a CDN?*. Fastly Documentation. Disponible en: <https://www.fastly.com/documentation/solutions/tutorials/introduction-to-cdn/1-introduction>. Último acceso: 22 de abril de 2025.
- [22] Fastly. (s.f.). *Working with CNAME records and your DNS provider*. Fastly Documentation. Disponible en: <https://docs.fastly.com/en/guides/working-with-cname-records-and-your-dns-provider>. Último acceso: 22 de abril de 2025.
- [23] Cloudflare. (s.f.). *What is a DNS PTR record?*. Disponible en: <https://www.cloudflare.com/learning/dns/dns-records/dns-ptr-record/>. Último acceso: 23 de abril de 2025.
- [24] Cisco. (2024). *Guide to troubleshooting Akamai-served content and domains after switching to Cisco Umbrella*. Disponible en: <https://support.umbrella.com/hc/en-us/articles/230563447-Guide-to-troubleshooting-Akamai-served-content-and-domains-after-switching-to-Cisco-Umbrella>. Último acceso: 23 de abril de 2025.
- [25] IBM. (s.f.). *IPv6 address formats*. IBM Documentation – IBM i 7.4. Disponible en: <https://www.ibm.com/docs/en/i/7.4.0?topic=concepts-ipv6-address-formats>. Último acceso: 24 de abril de 2025.
- [26] Cloudflare. (s.f.). *What is DNS?*. Cloudflare Learning Center. Disponible en: <https://www.cloudflare.com/learning/dns/what-is-dns/>. Último acceso: 24 de abril de 2025.