Tema 4. Servicio DNS (*Domain Name System*)

1. Introducción

El sistema de nombres de dominio DNS (*Domain Name System*) proporciona un mecanismo eficaz para llevar a cabo la resolución de nombres de dominio a direcciones IP. Como usuarios (humanos) nos es más fácil dirigirnos a un nombre de dominio (de host, de web, de servidor de correo, etc.) utilizando un texto identificativo (por ejemplo, `www.gva.es`) que a la dirección IP pertinente (por ejemplo, 193.144.127.85). el servicio DNS no sólo permite hacer la resolución de nombres de dominio a direcciones IP, sino también la resolución inversa. Es decir, a partir de una IP averiguar el nombre de dominio.

El servicio DNS proporciona independencia del nombre de dominio respecto a la IP. Así un dominio puede cambiar de IP de forma transparente para los usuarios del dominio. incluso es usual que un dominio se identifique con más de una IP como medida de redundancia contra la caída del sistema o como balanceo de cargas. Otros servicios proporcionados por el DNS son la identificación de los servidores de correo de un dominio, de cada uno de los hosts que pertenecen a la red, servidores de impresión, etc.

2. Sistemas de nombres planos y jerárquicos

El problema de la identificación de equipos se produce desde el principio de la existencia de las redes de ordenadores y no es algo específico de TCP/IP. Hacía falta un lenguaje humano para realizar esta identificación.

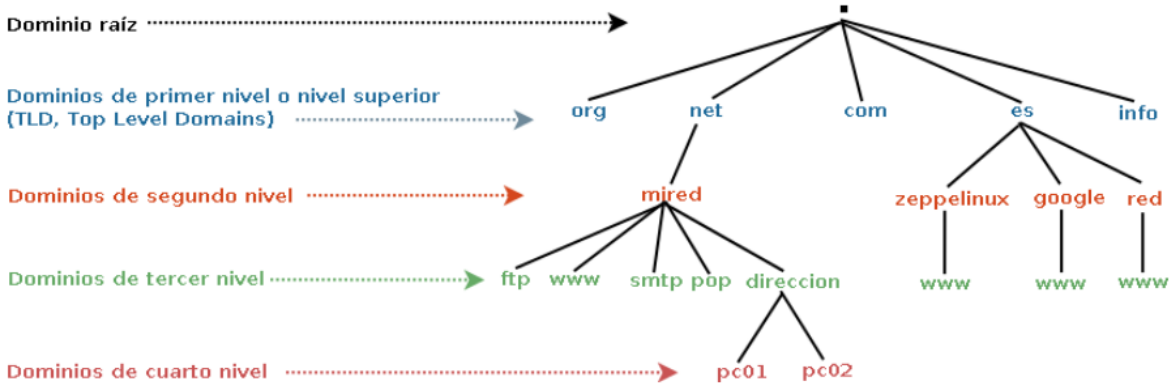
En los albores de las redes, cuando ARPANET (la red predecesora de Internet), los nombres los equipos se centralizaban en un archivo llamado `host.txt` (`/etc/hosts` en Linux), que incluía el nombre del equipo y su IP. Esto es lo que se conoce como un sistema de nombres plano. Puede ser adecuado para redes pequeñas, pero no es escalable ni práctico en redes grandes y mucho menos en Internet.

Ejemplo de fichero de nombres plano:

```
[root@portatil ~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1     localhost.localdomain   localhost   localhost
::1           localhost6.localdomain6 localhost6
192.168.1.1   router routerWF
192.168.1.31  server1   escriptori  pare
192.168.1.32  estacio1  dormitori   mare
192.168.1.33  estacio2  nen         jocs    supercrac
```

2.1 Elementos del sistema de nombres de dominio

El **espacio de nombres de dominio** está formado por los nombres válidos utilizados para identificar servicios o máquinas en una red. Se puede representar mediante una **estructura jerárquica de topología arbórea**, es decir, todos los nombres forman un árbol invertido donde cada nodo se separa de los otros nodos por un punto.



Nombres de dominio

Los nombres de dominio pueden estar formados por una o más cadenas de caracteres separadas por puntos y no se distingue entre mayúsculas y minúsculas. Por ejemplo, `www.deaw.es.` es lo mismo que `WWW.deaw.ES`.

Ejemplo de nombre de dominio: pc1.direccion.mired.net.

Los nombres de dominio se expresan como secuencias de **etiquetas (labels)**.

Dominios raíz

En teoría, todos los dominios deben de terminar con un punto (.). Es así porque el **árbol de nombres de dominio (espacio de nombres de dominio)** empieza con el dominio **.** que se conoce como **dominio raíz (*root*)**. En realidad, es un elemento nulo de 0 caracteres que se representa con un punto (.).

Un dominio se lee de derecha a izquierda, empezando por el punto, aunque en la práctica lo hacemos de izquierda a derecha. El punto inicial, generalmente se omite ya que los programas lo añaden por defecto y es meramente formal, pero en ocasiones, será necesario que indiquemos el nombre de dominio completo incluyendo el **dominio raíz**, es lo que se conoce como nombres de dominio completos (***Fully Qualified Domain Names*, FQDN**).

Dominios y subdominios

Como consecuencia de la organización jerárquica del espacio de nombres de dominios, podemos utilizar los términos dominio y subdominio. Por ejemplo, `deaw.es.` es un subdominio del dominio es. y `www.deaw.es.` es un subdominio del dominio `deaw.es.`

Los dominios o subdominios que cuelgan del dominio raíz se conocen como dominios de primer nivel o dominios de nivel superior (*Top Level Domains*, TLD), los que cuelgan de los dominios TLD se denominan dominios de segundo nivel y así sucesivamente.

2.2 Zonas

Una zona es una porción del espacio del espacio de nombre de dominio en el DNS cuya responsabilidad administrativa recae sobre un único responsable.

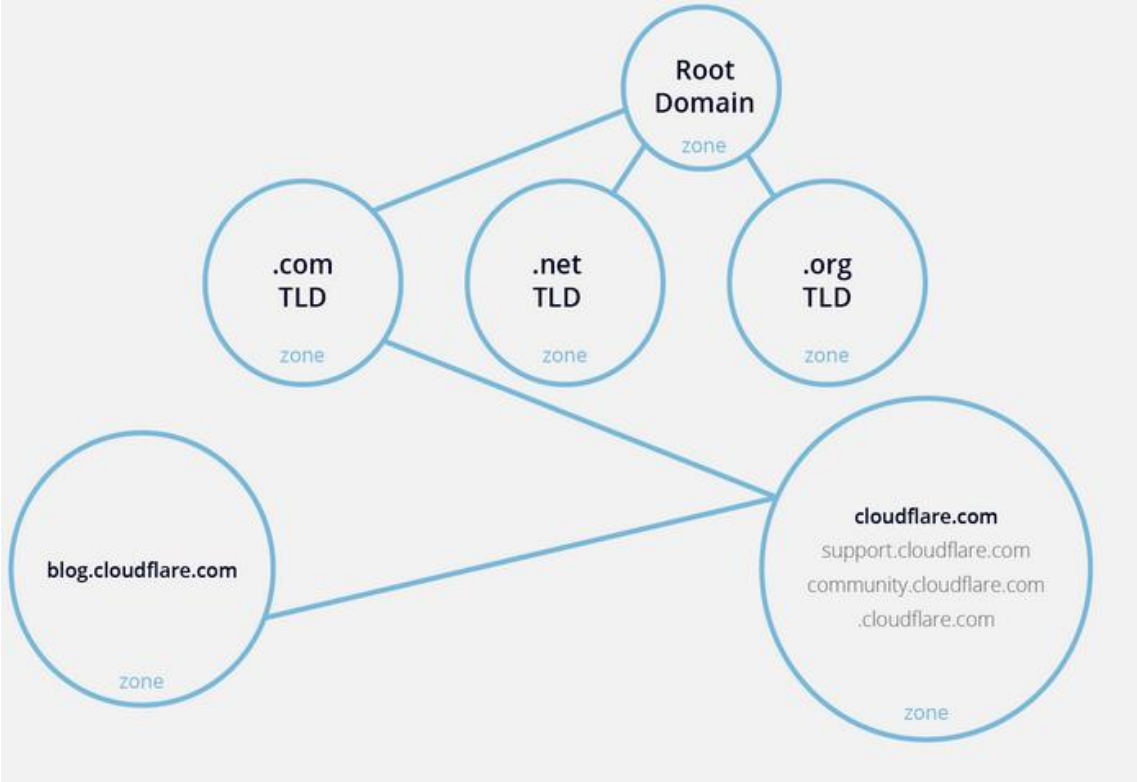
Los servidores que gestionan la zona tienen información completa sobre ella; y se dice que son autorizados para esa zona. Se les conoce como **servidores autoritativos**.

Las **zonas** se almacenan en **archivos de texto o en bases de datos**, según el tipo de software que se utilice para montar el servidor DNS y de cómo se configure. Tomemos como ejemplo el dominio deaw.es. y veamos parte de su archivo de zona

```
...
deaw.es.          IN NS      ns1.deaw.es.
ns1.deaw.es.      IN A       192.168.1.20
goku.deaw.es.     IN A       192.168.1.21
luffy.deaw.es.    IN A       192.168.1.22
www.deaw.es.      IN CNAME   goku.deaw.es.
ftp.deaw.es.      IN CNAME   luffy.deaw.es.
...
```

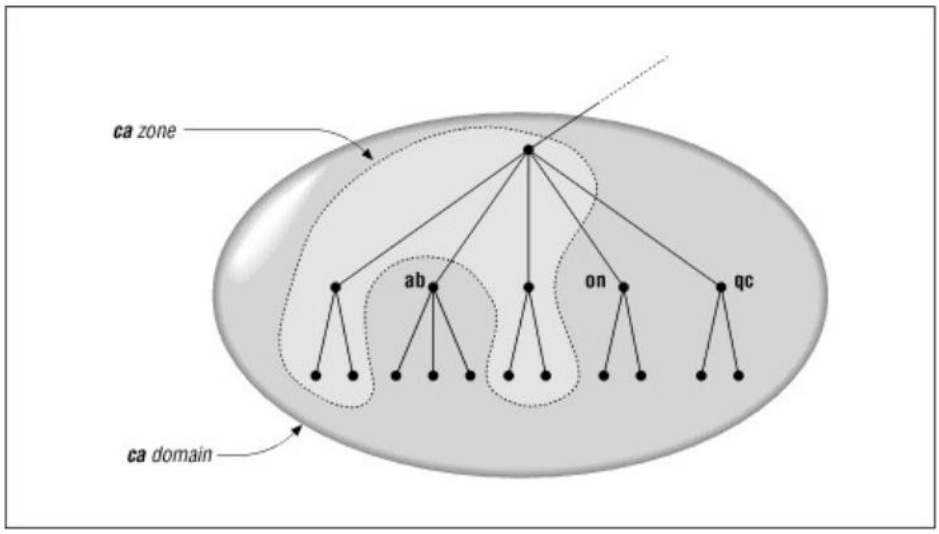
A cada una de las líneas del fichero se las conoce como registros de recurso (*RR: Resource Records*) y definen los tipos de datos en el *Domain Name System* (DNS). Se utilizan para almacenar datos sobre nombres de dominio y direcciones IP. Una base de datos o fichero de zona está formada por una serie de registros de recursos. Cada registro de recurso da información pertinente sobre un objeto determinado. Por ejemplo, los **registros de tipo (A)** asocian un nombre de host con una dirección IP, y los **registros de puntero de búsqueda inversa (PTR)** asocian una dirección IP con un nombre de host y un **registro (NS, *Name Server***) define un servidor DNS para la zona. El servidor DNS utiliza estos registros de recurso para resolver las consultas de los hosts de su zona.

Cuando un servidor DNS **es autorizado** para una zona, es el responsable de los nombres de dominio para esa zona. En nuestro ejemplo, `ns1.deaw.es` es el servidor autorizado para la zona `deaw.es`. y en él se definen los nombres que cuelgan de `deaw.es` como, por ejemplo, `www.deaw.es`, `ftp.deaw.es`, `goku.deaw.es`, etc.



La organización que administra el servidor DNS y por lo tanto la zona, puede delegar o no alguno de sus subdominios. Supongamos que de `deaw.es`. cuelgan los subdominios `teoria.deaw.es.` y `practicas.deaw.es.` y se decide delegar solo el subdominio `practicas.deaw.es.` . Esto implica que existirá otro servidor DNS autorizado para el dominio `practicas.deaw.es.` , que almacenará el fichero de zona para dicho dominio.

Una zona no es lo mismo que un dominio. Un **dominio** es un **subarbol del espacio de nombres de dominio** y los datos asociados a los nombres de un **dominio** pueden estar almacenados en una o varias **zonas**, distribuidas en uno o varios **servidores DNS**. Básicamente una zona es una porción de dominio.



Tipos de RR (*Resource Record*)

En esta subsección vamos a ver cuáles son los registros de recursos o RR más utilizados. Antes debemos aclarar algunos conceptos:

`$TTL (TIME TO LIVE)`

El TTL o tiempo de vida determina durante cuánto tiempo son válidos los RR. Pueden indicarse en semanas (`$TTL 1W`), días (`$TTL 7D`), horas (`$TTL 168H`) o minutos (`$TTL 10080M`).

En otras palabras, el TTL indica cuánto tiempo tardarán en aplicarse los cambios que le hagamos a un RR desde que los hacemos. En el ejemplo del párrafo anterior, los servidores DNS comprobarán cada semana si se ha producido algún cambio en esos RR. **Debe declararse al inicio del archivo de zona**.

`$ORIGIN`

La directiva `$ORIGIN` define el nombre del dominio que será **añadido** al final de cualquier nombre que no acabe en punto (nombres relativos o no cualificados) en los RR, para así transformarlos en nombres FQDN (*fully qualified domain name*). Si un nombre acaba en punto, se considera un nombre FQDN y no se utilizaría `$ORIGIN`.

Su sintaxis o forma de escribirlo será:

```
$ORIGIN nombre-dominio
```

Por ejemplo:

```
$ORIGIN deaw.es.

;A partir de aquí se añade deaw.es. a todos los nombres relativos
```

FORMATO GENERAL DE LOS RR

El formato con el que se introducen los RR en los archivos de zona es del siguiente estilo:

```
Nombre de dominio       [TTL]       Clase       Tipo       Tipo-Dato
```

Así, por ejemplo, un RR escrito de forma **absoluta** quedaría tal que así:

```
profesor.deaw.es.     7200        IN      A       192.168.10.254
```

Y, el mismo RR podría escribirse así de forma **relativa**, si su `$ORIGIN` es `deaw.es.` :

```
profesor     7200        IN      A       192.168.10.254
```

TIPOS DE REGISTROS

- **Registro SOA (*Start Of Authority*)**: Especifica información autoritaria sobre una zona DNS, incluyendo el servidor de nombre primario, el email del administrador, el número de serial o versión de la zona, y varios temporizadores.

  Ejemplo:

```
deaw.es.    IN    SOA    ns1.deaw.es.    super.deaw.es. (
                  20250425001 ; serial
                  7D  ; refresh (7 días)
                  1D   ; retry (1 día)
                  4W ; expire (28 días)
                  1W )    ; TTL negativo (7 días)
```

CAMPO	SIGNIFICADO	EN EL EJEMPLO
DEAW.ES.	Nombre de la zona (dominio raíz para este archivo)	`deaw.es.`
IN	Clase de registro DNS. Siempre es IN (Internet)	`IN`
SOA	Tipo de registro. Indica el inicio de autoridad de zona	`SOA`
NS1.DEAW.ES.	Servidor DNS primario autorizado para esta zona.	`ns1.deaw.es.`
SUPER.DEAW.ES	Email del administrador de la zona. El primer . se lee como @	`super@deaw.es`

20250425001	**Serial**: número de versión de la zona. Los esclavos lo usan para saber si deben actualizarse.	`20190425001` (formato `YYYYMMDDnn` habitual)
7D	**Refresh:** tiempo que un servidor esclavo espera antes de consultar al primario si hay cambios.	7 días
1D	**Retry:** si falla el refresh, tiempo que espera el esclavo antes de volver a intentarlo.	1 día
4W	**Expire:** tiempo máximo que el esclavo seguirá usando la zona si no logra contactar al primario.	28 días
1W	**Negative TTL**: tiempo que un servidor cacheará respuestas negativas (cuando un registro no existe).	7 días

- **Registro NS (*Name Server*)**: indica **qué servidor(es) DNS son autoritativos** para una zona o subzona de dominio.
  - En la **zona raíz** (.), los NS apuntan a los servidores raíz
  - En la **zona de un dominio** (`deaw.es.`), los NS definen **qué servidores tienen autoridad** sobre todo `deaw.es.`
  - En una **delegación de subdominio** (ej: `practicas.deaw.es.`), los NS indican **qué servidores gestionan esa subzona**.

```
...
deaw.es.     IN  NS  ns1.deaw.es.     ;Servidor DNS maestro
deaw.es.     IN  NS  ns2.deaw.es.     ;Servidor DNS esclavo
deaw.es.     IN  NS  dns.deaw.net.    ;Servidor DNS esclavo

ns1.deaw.es.     IN  A   192.168.10.20
ns2.deaw.es.     IN  A   192.168.10.21

;DELEGACIÓN
practicas.deaw.es.  IN  NS  ns1.practicas.deaw.es.
redes.deaw.es.  IN  NS  dns.deaw.net.
```

- **El regsitro A (*Address*)**: también conocido como registro de dirección, establece una correspondencia entre un nombre de dominio completamente cualificado (FQDN) y una dirección IP versión 4.

```
...
ns1.deaw.es.          IN  A    192.168.10.20
ns2.deaw.es.          IN  A    192.168.10.21
natos.deaw.es.        IN  A    192.168.10.22
...
```

- **El registro CNAME (*Canonical Name*)** permite crear alias para nombres de dominio ya especificados en registros A.

```
...
goku.deaw.es.         IN  A    192.168.1.22
www.deaw.es.          IN  CNAME    goku.deaw.es.
ftp.deaw.es.          IN  CNAME    goku.deaw.es.
...
```

Un registro CNAME también puede apuntar a otro nombre de dominio:

```
...
www.deaw.es.     IN  CNAME    www.deaw.com.
...
```

- **El registro MX (*Mail Exchange*)** permite definir los servidores encargados de la recepción/entrada de correo en el dominio (servidores SMTP) y la prioridad (cuanto menor valor más alta es la prioridad) entre ellos. Su sintaxis es la siguiente:

```
...
deaw.es.     IN  MX  10  mail1.deaw.es.
deaw.es.     IN  MX  20  mail2.deaw.es.

mail1.deaw.es.  IN  A   192.168.1.100
mail2.deaw.es.  IN  A   192.168.1.101
...
```

En tu ejemplo, `deaw.es` tiene dos servidores de correo: `mail1.deaw.es` con prioridad **10** y `mail2.deaw.es` con prioridad **20**. Eso significa que los correos que se envíen a alguien del dominio `deaw.es` (por ejemplo, `usuario@deaw.es`) se intentarán entregar primero a mail1 (192.168.1.100) y, si este no responde, se usaría mail2 (192.168.1.101) como respaldo.

- **El registro PTR (*Pointer Record*)** establece una correspondencia entre direcciones IPv4 e IPv6 y nombres de dominio. Se utilizan en las zonas de resolución inversa.
  En el caso de un bloque IPv4 de prefijo `/24`, por ejemplo, el `192.168.1.0/24`, los registros PTR serían los siguientes:

```
...
20.1.168.192.in-addr.arpa.    IN    PTR    ns1.deaw.es.
21.1.168.192.in-addr.arpa.    IN    PTR    ns2.deaw.es.
22.1.168.192.in-addr.arpa.    IN    PTR    goku.deaw.es.
...
```

  O lo que es lo mismo:

```
...
20    IN    PTR    ns1.deaw.es.
21    IN    PTR    ns2.deaw.es.
22    IN    PTR    natos.deaw.es.
...
```

- **El registro TXT (*plaint text*)** es un tipo de registro DNS que permite asociar **información arbitraria en forma de texto** a un dominio. Originalmente se pensó como un campo genérico para notas o descripciones, pero hoy en día tiene usos muy importantes en servicios de Internet:
  - ***Verificación de dominios.*** Muchos servicios (Google, Microsoft, AWS…) te piden añadir un TXT para demostrar que eres el dueño del dominio.

```
@   IN  TXT   "google-site-verification=abc123"
```

  - ***Políticas de correo (seguridad).***
    - SPF (*Sender Policy Framework*): define qué servidores pueden enviar correos en nombre de tu dominio.

```
@   IN  TXT   "v=spf1 ip4:192.168.1.100 -all"
```

    - DKIM (*DomainKeys Identified Mail*): almacena claves públicas para firmar correos.
    - DMARC (*Domain-based Message Authentication, Reporting & Conformance*): políticas adicionales de autenticación de correo.
  - **Información arbitratia**. Guardar cualquier cadena de texto.

```
...
@   IN  TXT     "Servidor maestro de DEAW"
```

3. Tipos de servidores DNS

3.1 Servidor maestro o primario

Un servidor maestro o primario, define una o varias zonas de las que es autorizado. Sus archivos de zona son de lectura y escritura y es en ellos donde el administrador del servidor añade, modifica o elimina nombres de dominio.

- Si un cliente DNS u otro servidor DNS le pregunta por algún nombre de dominio **para el que es autorizado**, consulta con los ficheros de zona y responde a la pregunta. **A esto se lo denomina respuesta autoritativa.**
- Si un cliente DNS u otro servidor DNS le pregunta por algún nombre de dominio para el que **no es autorizado**, tendrá que preguntar a otros servidores DNS o responder que no conoce la respuesta. También es posible que tuviera en la caché la respuesta almacenada (no en el archivo de zona), lo que se denominaría como **respuesta caché.**

3.2 Servidor esclavo o secundario

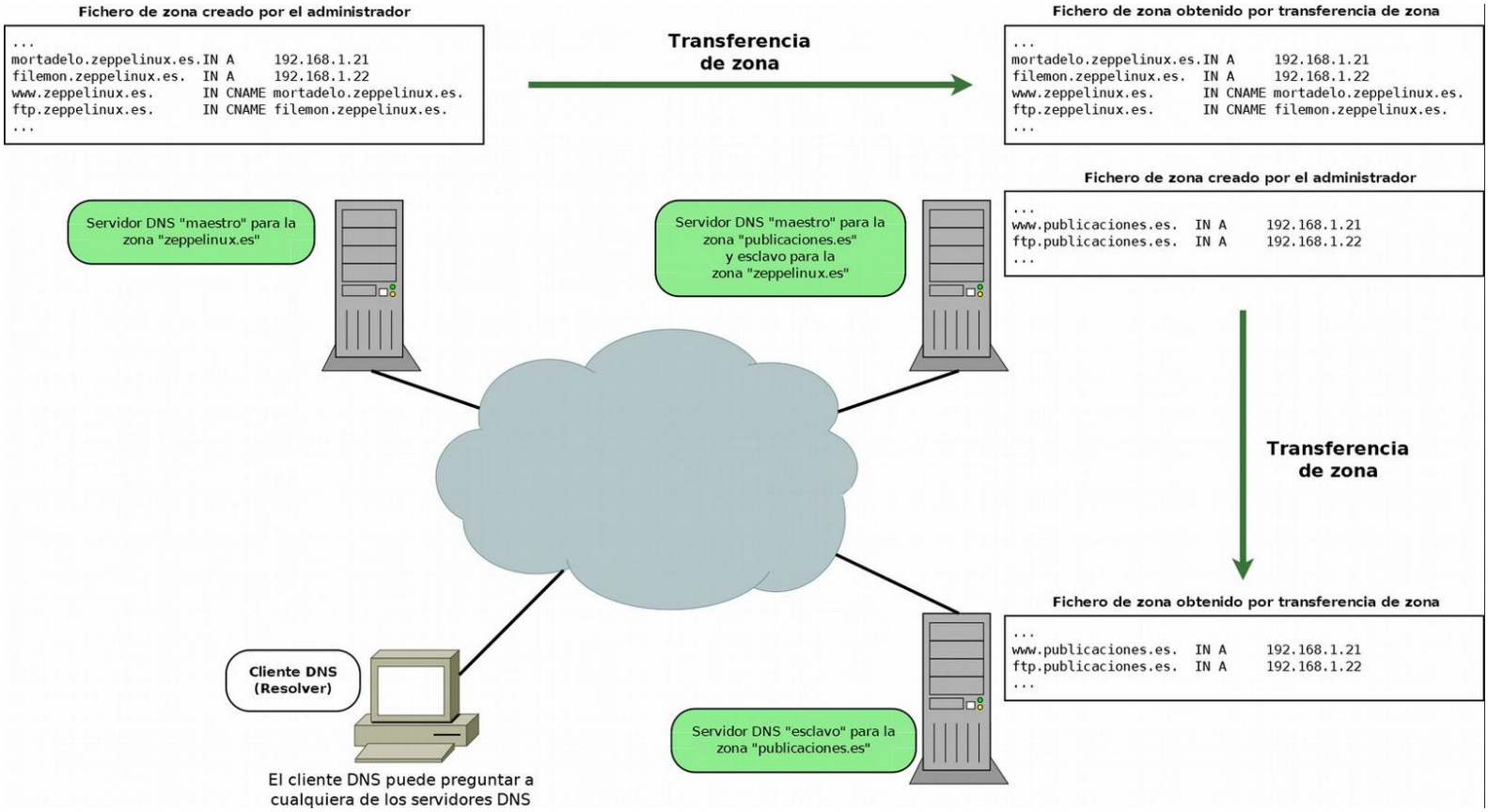
Un **servidor esclavo o secundario** define una o varias zonas para las que **es autorizado**. La diferencia con respecto a un servidor maestro es que **los ficheros de zona los obtiene de otro servidor autorizado para la zona**, normalmente, de un servidor maestro mediante un procedimiento denominado **transferencia de zona**. Los ficheros de zona de los servidores esclavos son de solo lectura y, por lo tanto, el administrador no tiene que editarlos. La modificación de los archivos de zona debe realizarla el servidor maestro que transfiere la zona.

El funcionamiento de cómo responden a los clientes DNS o a otros servidores DNS es similar al de un servidor maestro. Un servidor puede ser maestro para una o varias zonas y al mismo tiempo ser esclavo para otras.

Pueden existir varios servidores esclavos para una misma zona. Las razones para esto suelen ser:

- Reducir y repartir la carga entre varios servidores DNS.
- Favorecer la tolerancia a fallos.
- Ofrecer mayor rapidez.

Lo ideal es que los servidores DNS para una misma zona estén ubicados en redes y localizaciones diferentes para evitar que, si ocurre algún problema no les afecte simultáneamente y deje sin servicio de resolución a los nombres de esa zona.



3.3 Servidor caché

Los servidores DNS también se pueden configurar como servidores caché para mejorar los tiempos de respuesta de las consultas, reducir la carga de los equipos y disminuir el tráfico de red.

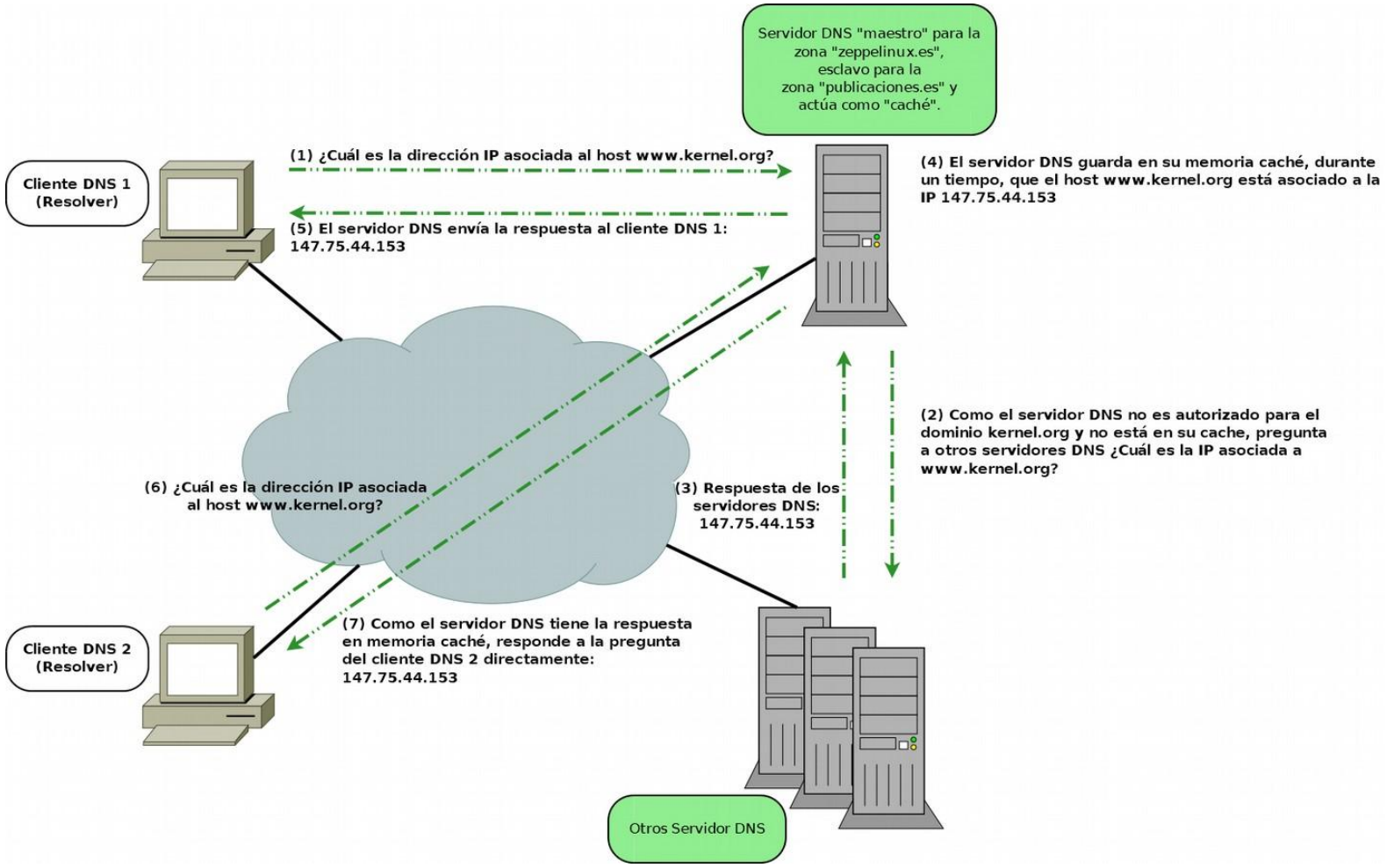
Cuando un servidor DNS recibe una *request* sobre un dominio para el cual no es autorizado, es decir, de un nombre del cual no tiene información, puede preguntar, si así está configurado, a otros servidores para obtener la respuesta. Si el servidor actúa como caché, guarda durante un tiempo (TTL: *Time To Live*) las respuestas a las últimas preguntas que ha realizado a otros servidores DNS. Cada vez que un cliente DNS u otro

servidor DNS le formula una pregunta, comprueba si tiene la respuesta en su memoria caché y, si la tiene, no tendrá que preguntar a otro servidor DNS por dicho dominio.

Un servidor DNS es solo cache (*cache only server*) cuando:

- No tiene autoridad sobre ninguna zona.
- Pregunta a otros servidores DNS para resolver las preguntas de los clientes DNS y las guarda en su memoria cache.

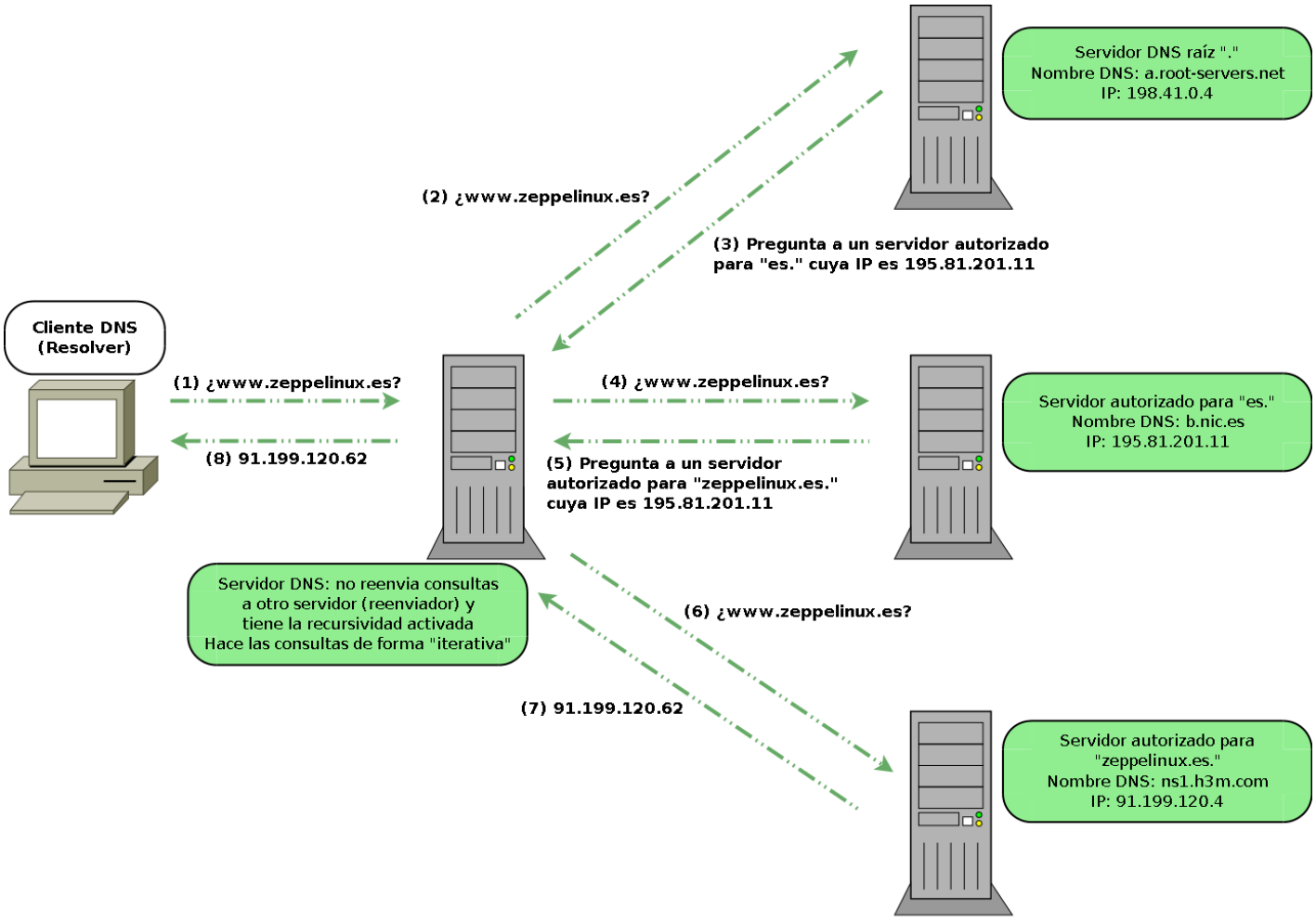
En el siguiente gráfico se explica como dos clientes DNS hacen preguntas a un mismo servidor DNS que es autorizado para algunas zonas y además actúa como caché.



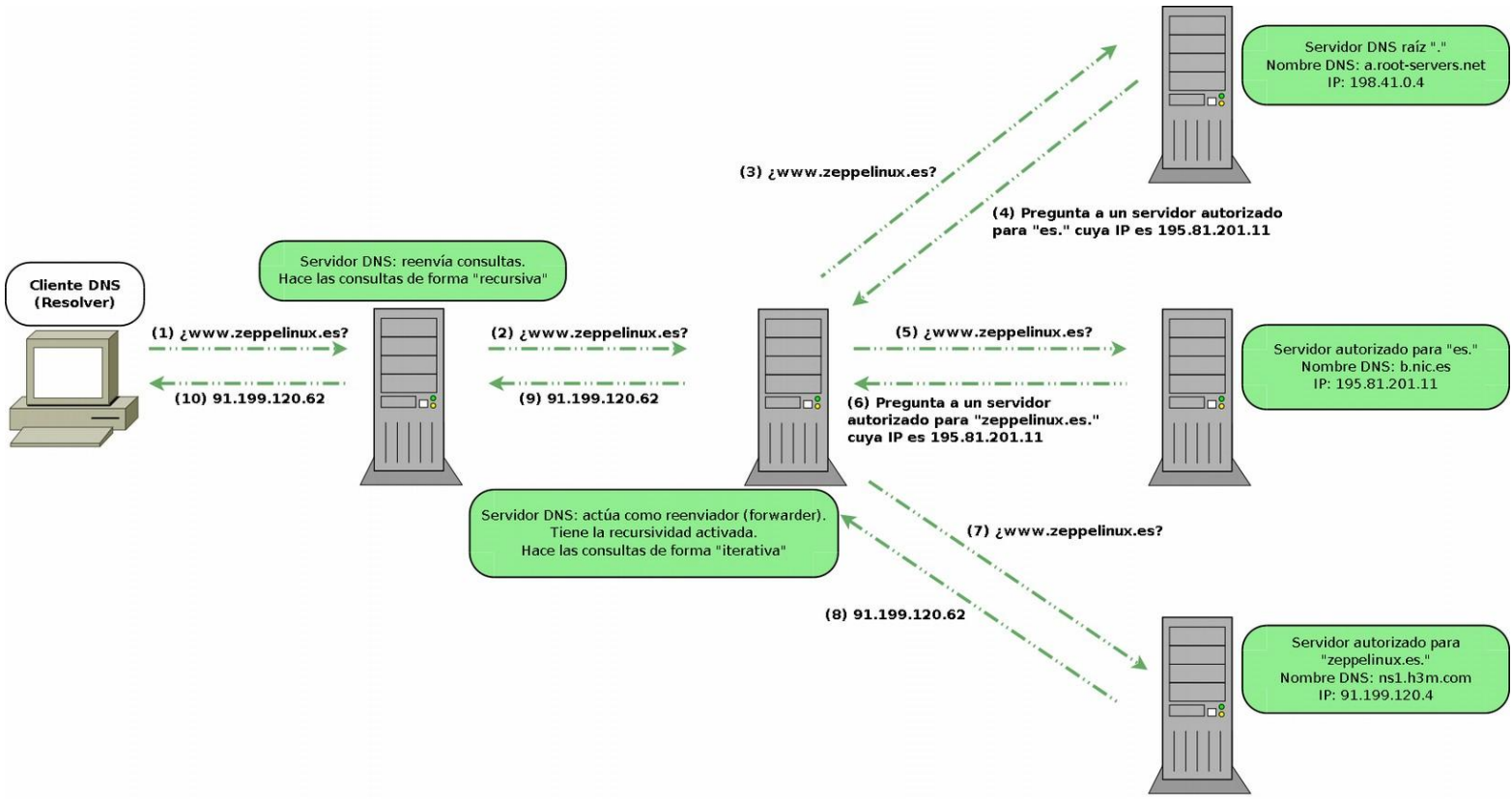
3.4 Servidores

Cuando a un servidor DNS se le hace una pregunta sobre un nombre de dominio del que no dispone información (no es autorizado), este puede preguntar a otros servidores DNS. Y, existen dos formas de procesar las consultas:

- **Consulta iterativa**. Va preguntando paso a paso, empezando por los servidores raíz, luego los TLD (.com, .es, etc.), después los autoritativos del dominio… hasta llegar al que sabe la respuesta. En cada paso, el servidor consultado **no busca más allá**, solo responde con un *referral*, es decir, una referencia con la dirección de otros servidores más cercanos a la respuesta". El *resolver* es el que hace todo el trabajo de ir saltando de servidor en servidor.



- **Consulta recursiva**. El servidor DNS al que preguntas (por ejemplo, el de tu ISP) se encarga él mismo de hacer las iteraciones. Tú como cliente solo haces **una consulta**, y ese servidor se encarga de preguntar a los raíz[1], TLD, autoritativos, etc., y luego te devuelve la respuesta final.



En nuestra práctica, si el servidor DNS tiene la **recursión activada**, entonces él mismo hace esas **consultas iterativas** (a raíz[2], TLD, autoritativos) y entrega la respuesta final al cliente.

En este caso, el funcionamiento con recursión es:

1. El **cliente** (por ejemplo, tu PC) le pide al servidor DNS: *"Dame la IP de www.ejemplo.com"*
2. Como el servidor **no es autoritativo** para ese dominio y no tiene la respuesta en caché, necesita buscarla.
3. Para conseguirla, el servidor **realiza internamente varias consultas iterativas**:

[1] Los servidores DNS tienen un listado de direcciones IP de los servidores raíz, se denominan ***sugerencias raíz***. En BIND están en el fichero `/etc/bind/db.root` o compilados en el propio software.

1. Pregunta a un servidor raíz → recibe un *referral* a los servidores del TLD `.com`
2. Pregunta a un servidor de `.com` → recibe un *referral* a los servidores autoritativos de `ejemplo.com`.
3. Pregunta a un servidor autoritativo de `ejemplo.com` → recibe la respuesta final (la IP de `www.ejemplo.com`).

Además, en lugar de que nuestro servidor recursivo consulte directamente a los servidores raíz, podemos configurarlo para que utilice ***forwarders* (reenviadores);** de hecho, en caso de que existan ambos métodos, **por defecto** el servidor DNS intentará primero resolver la consulta mediante *forwarders*. Un *forwarder* es otro servidor DNS (por ejemplo, el de nuestro ISP o uno público como 8.8.8.8 de Google o 1.1.1.1 de Cloudflare) al que nuestro servidor envía las consultas externas que no sabe resolver.

En este caso, el funcionamiento con *forwarders* sería:

1. El cliente de nuestra red hace una consulta a nuestro servidor DNS.
2. Nuestro servidor, al no ser autoritativo para ese dominio, reenvía la consulta al *forwarder* configurado.
3. El *forwarder* se encarga de hacer todas las iteraciones necesarias (raíz, TLD, autoritativos) y devuelve la respuesta completa.
4. Finalmente, nuestro servidor entrega esa respuesta al cliente como si él mismo la hubiera resuelto.

Esto tiene varias ventajas:

- Reduce el tráfico hacia los servidores raíz, ya que solo el *forwarder* realiza esas consultas.
- Aprovecha la **caché** del *forwarder*, acelerando respuestas repetidas.
- Permite **centralizar y controlar** la resolución de nombres (por ejemplo, aplicando filtros de seguridad o políticas de acceso).

En BIND, la configuración típica de *forwarders* es:

```
options {
    forwarders {
        8.8.8.8;    // Google DNS
        1.1.1.1;    // Cloudflare DNS
    };
    forward only;
};
```

3.5 Servidor sólo autorizado

Un Servidor solo autorizado (*authoritative only*) es aquel que es autorizado para una o varias zonas como servidor maestro y/o esclavo y **no responde a preguntas que no sean relativas a sus zonas**. Es decir, no tiene activada la recursividad, no es reenviador y no actúa como caché.

3.6 Servidores raíz

En Internet existen un conjunto de servidores DNS autorizados para el dominio raíz **.** , conocidos como servidores raíz (*root servers*). Contienen el fichero de la zona **.** que contiene información sobre los servidores DNS autorizados para cada uno de los dominios TLD.

Los servidores raíz son una parte fundamental de Internet, son el primer paso en la traducción (resolución) de los nombres de host en direcciones IP, que se utilizan en la comunicación entre los hosts de Internet. Son claves en el proceso de resolución de nombres de dominio en Internet, y deben de ser conocidos por todos los servidores DNS que respondan a preguntas sobre nombres para los que no son autorizados.

Existen 13 servidores raíz en toda Internet y cada uno de ellos tiene múltiples copias distribuidas por todo el mundo, es decir, que físicamente no solo son 13 servidores. Cada conjunto de copias de uno de los 13 servidores se identifica por una misma IP. Cuando un cliente realiza una pregunta a una IP de un servidor raíz, los *routers* de Internet encaminan la pregunta hacia la copia más cercana mediante un procedimiento denominado *anycasting*.

Los nombres de los servidores raíz son de la forma letra.root-servers.net, donde letra va desde la A a la M. Esta página permite ver donde están ubicados y replicados.

```
;================================================================================================
;Hostname               Dirección IP                                Administrador
;================================================================================================
a.root-servers.net      198.41.0.4, 2001:503:ba3e::2:30             VeriSign, Inc.
b.root-servers.net      199.9.14.201, 2001:500:200::b               University of Southern Californ
c.root-servers.net      192.33.4.12, 2001:500:2::c                  Cogent Communications
d.root-servers.net      199.7.91.13, 2001:500:2d::d                 University of Maryland
e.root-servers.net      192.203.230.10, 2001:500:a8::e              NASA (Ames Research Center)
f.root-servers.net      192.5.5.241, 2001:500:2f::f                 Internet Systems Consortium, In
g.root-servers.net      192.112.36.4, 2001:500:12::d0d              US Department of Defense (NIC)
h.root-servers.net      198.97.190.53, 2001:500:1::53               US Army (Research Lab)
i.root-servers.net      192.36.148.17, 2001:7fe::53                 Netnod
j.root-servers.net      192.58.128.30, 2001:503:c27::2:30           VeriSign, Inc.
k.root-servers.net      193.0.14.129, 2001:7fd::1                   RIPE NCC
l.root-servers.net      199.7.83.42, 2001:500:9f::42                ICANN
m.root-servers.net      202.12.27.33, 2001:dc3::35                  WIDE Project
;================================================================================================
```

4. Tipos de consultas: recursivas e iterativas

4.1 Consultas recursivas

Una consulta recursiva es aquella en la que el servidor DNS se compromete a darte una **respuesta final**. Pueden darse tres tipos de respuesta:

- *Positivas*: encuentra el registro y devuelve la información (ej: IP del dominio).
- *Negativas*: confirma que el dominio no existe (`NXDOMAIN`).
- *Error*: hubo un fallo de red, de configuración o de tiempo de espera (ej: `SERVFAIL`).

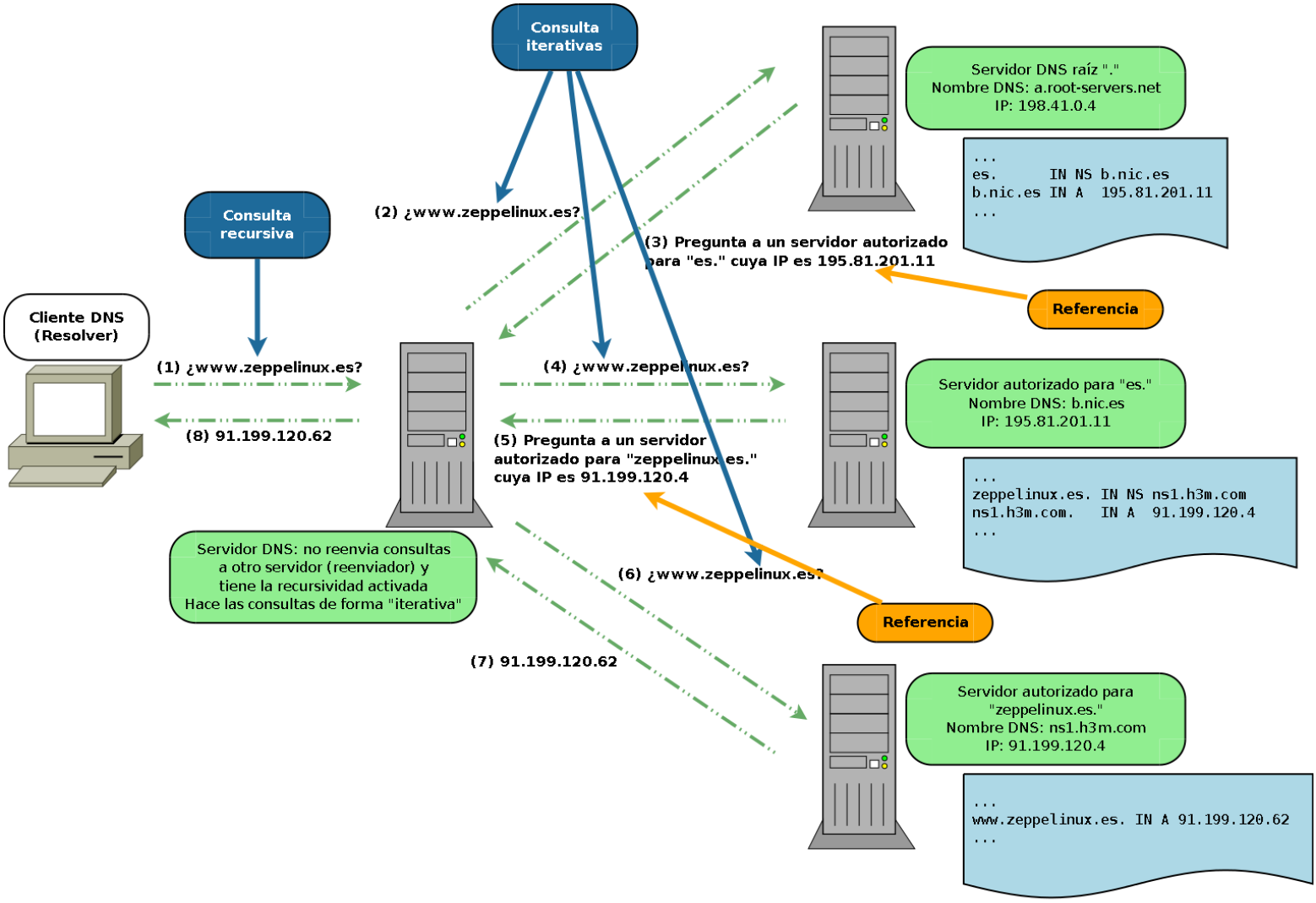
4.2 Consultas iterativas

El servidor DNS no resuelve todo por ti, sino que responde solo con lo que sabe. Pueden darse 4 tipos de respuesta:

- *Positiva*: tiene el registro y lo devuelve.
- *Negativa*: sabe que el dominio no existe en esa zona
- *Referral*: te da la dirección de otros servidores que están más cerca de la respuesta
- *Error*: hay un fallo de red, de configuración o de tiempo de espera.

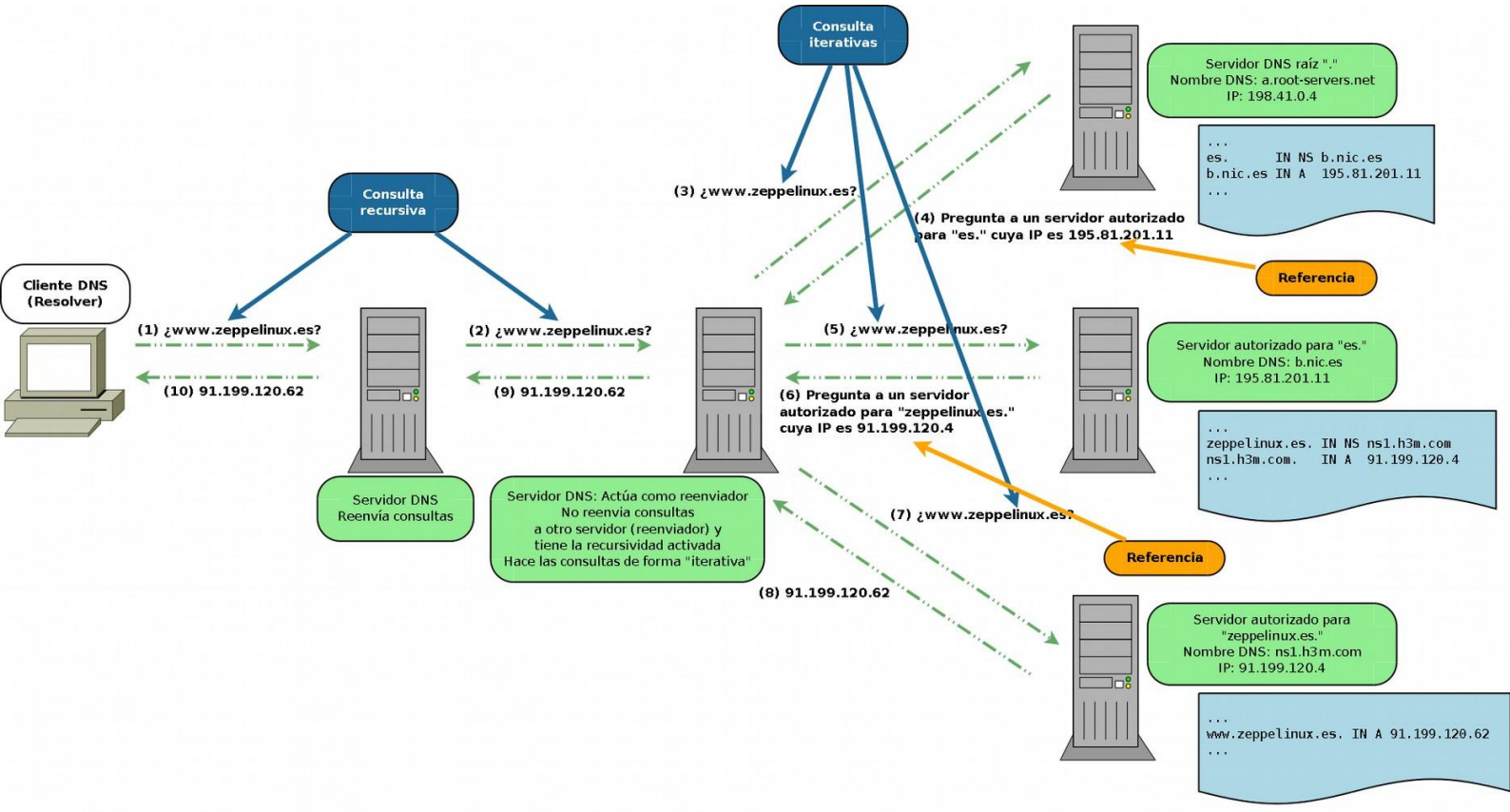
EJEMPLOS

Completando la información de la imagen del primer ejemplo del apartado del **reenviador *forwarder***:



Ejemplo de resolución DNS nº 1

Completando la información de la imagen del segundo ejemplo del apartado del **reenviador *forwarder***:



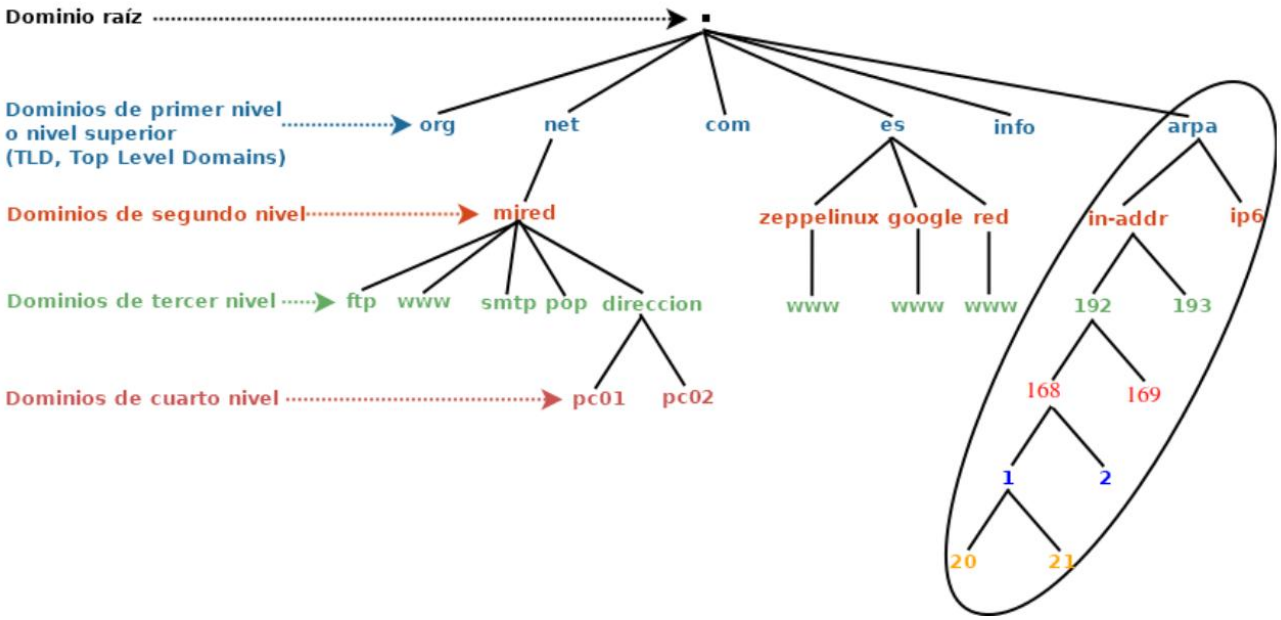
Ejemplo de resolución DNS nº 2

5. Resolución inversa

La resolución inversa consiste en obtener información de un nombre de dominio preguntando por la dirección IP en vez de preguntar por el nombre de domino como hemos explicado en apartados anteriores.

5.1 Mapeo de direcciones y el dominio arpa

El funcionamiento de la resolución de direcciones IP es igual al de la resolución de nombres de dominio. Las direcciones IP se tratan como nombres que cuelgan del dominio ***in-addr.arpa*** para las direcciones IPv4, y del dominio ***ip6.arpa*** para las direcciones IPv6.



Cuando usamos una dirección IP, por ejemplo `192.168.1.21`, para realizar una pregunta DNS inversa, en realidad estamos preguntando por el nombre de dominio **`21.1.168.192.in-addr.arpa`** . La estructura jerárquica de la dirección IP, tratada como nombre de dominio, es de derecha a izquierda, comenzando por el dominio `in-addr.arpa.`

`.arpa` (*Address and Routing Parameter Area*) es un dominio de nivel superior genérico utilizado sólo para la infraestructura de Internet. Los subdominios de `.arpa` o dominios de segundo nivel «`in-addr.arpa`» e «`ip6.arpa`» son usados por los servidores DNS inversos para la obtención de direcciones IPv4 e IPv6 respectivamente.

Cuando mapeamos una dirección IP estamos asociando la dirección IP al nombre en el dominio `.arpa`. Por ejemplo, la dirección `192.168.1.21` es mapeada al nombre **`21.1.168.192.in-addr.arpa`**.

5.2 Zonas de resolución inversa

Los servidores DNS almacenan zonas de resolución inversa con registros de recursos (RR) que asocien nombres de dominio con direcciones IP. Las zonas de resolución inversa pueden ser maestras/primarias y esclavas/secundarias.

Las zonas de resolución directa e inversa son independientes y es responsabilidad de los administradores de los servidores DNS que dichas zonas contengan información coherente y que no existan discrepancias.

No es obligatorio que la entidad que administra una zona de resolución directa de un dominio tenga que administrar la zona de resolución inversa que se corresponda con las direcciones IP asociadas a dicho dominio.

Si recordamos la zona de resolución directa de `deaw.es`:

```
...
deaw.es.            IN  NS      ns1.deaw.es.
ns1.deaw.es.        IN  A       192.168.1.20
goku.deaw.es.       IN  A       192.168.1.21
luffy.deaw.es.      IN  A       192.168.1.22
altea.deaw.es.      IN  A       192.168.1.23
www.deaw.es.        IN  CNAME   goku.deaw.es.
ftp.deaw.es.        IN  CNAME   luffy.deaw.es.
...
```

Este sería el archivo de la zona de resolución inversa `1.168.192.in-addr.arpa` usando el **dominio absoluto,** el cual permite resolver consultas inversas sobre direcciones IP de la red `192.168.1.0/24`:

```
...
1.168.192.in-addr.arpa.     IN  NS  ns1.deaw.es.
20.1.168.192.in-addr.arpa.  IN  PTR ns1.deaw.es.
21.1.168.192.in-addr.arpa.  IN  PTR goku.deaw.es.
22.1.168.192.in-addr.arpa.  IN  PTR luffy.deaw.es.
23.1.168.192.in-addr.arpa. IN  PTR altea.deaw.es.
...
```

Y esto sería la zona inversa usando el **dominio relativo** (ambas son correctas):

```
…
@       IN  NS  ns1.deaw.es.
20      IN  PTR ns1.deaw.es.
21      IN  PTR goku.deaw.es.
22      IN  PTR luffy.deaw.es.
23      IN  PTR altea.deaw.es.
…
```

5.3 Proceso de resolución

El proceso de resolución inversa es similar al de resolución directa. Las direcciones IP se tratan como nombres de dominio. Por lo tanto, existen consultas recursivas, iterativas, cache, TTL...

Por ejemplo, si un cliente DNS realiza una consulta recursiva de la IP 192.168.1.21 a un servidor DNS, éste, si no lo tiene en cache, iniciará una serie de consultas iterativas a los servidores DNS raíz, a los servidores autorizados para el dominio `192.in-addr.arpa` y así sucesivamente.

6. Herramientas

6.1 Nslookup

Es un programa para consultar servidores DNS. Se utiliza para saber si un servidor DNS resuelve correctamente los nombres DNS y las direcciones IP, para solucionar problemas frecuentes de los servidores DNS o, para diagnosticar problemas ocasionales de configuración en los servidores DNS.

Con *nslookup* podemos obtener la dirección IP asociada a un nombre DNS y viceversa, además, podemos preguntar a los servidores de nombres información relativa a los registros de recursos (RR) de la/s zona/s de las que son autorizados.

nslookup se usa de **dos modos**: interactivo y no interactivo. El modo interactivo permite al usuario consultar los servidores DNS para obtener información sobre varios hosts y dominios o para listar los hosts de un dominio. El modo no interactivo se usa para presentar solo el nombre y la información solicitada para un host o nombre DNS.

Este comando funciona tanto en sistemas operativos UNIX/Linux como en Windows. En su momento se trató a *nslookup* como una aplicación "*deprecated*" u obsoleta, pero hoy en día parece que ha vuelto a considerarse apta para su uso normal.

```
C:\Users\Sergio>nslookup cisco.com
Servidor:  250.red-80-58-61.staticip.rima-tde.net
Address:  80.58.61.250

Respuesta no autoritativa:
Nombre:  cisco.com
Addresses:  2001:420:1101:1::185
          72.163.4.185
```

```
C:\Users\Sergio>nslookup -type=ns cisco.com
Servidor:  250.red-80-58-61.staticip.rima-tde.net
Address:  80.58.61.250

Respuesta no autoritativa:
cisco.com       nameserver = ns3.cisco.com
cisco.com       nameserver = ns1.cisco.com
cisco.com       nameserver = a3-64.akam.net
cisco.com       nameserver = a28-64.akam.net
cisco.com       nameserver = ns2.cisco.com

ns1.cisco.com   internet address = 72.163.5.201
a3-64.akam.net  internet address = 96.7.49.64
a3-64.akam.net  AAAA IPv6 address = 2600:1408:1c::40
a28-64.akam.net internet address = 95.100.173.64
a28-64.akam.net AAAA IPv6 address = 2600:1480:d800::40
ns2.cisco.com   internet address = 64.102.255.44
```

```
C:\Users\Sergio>nslookup
Servidor predeterminado:  250.red-80-58-61.staticip.rima-tde.net
Address:  80.58.61.250

> set type=ns
> cisco.com
Servidor:  250.red-80-58-61.staticip.rima-tde.net
Address:  80.58.61.250

Respuesta no autoritativa:
cisco.com       nameserver = ns3.cisco.com
cisco.com       nameserver = a3-64.akam.net
cisco.com       nameserver = ns1.cisco.com
cisco.com       nameserver = a28-64.akam.net
cisco.com       nameserver = ns2.cisco.com

a28-64.akam.net internet address = 95.100.173.64
a28-64.akam.net AAAA IPv6 address = 2600:1480:d800::40
ns2.cisco.com   internet address = 64.102.255.44
a3-64.akam.net  internet address = 96.7.49.64
a3-64.akam.net  AAAA IPv6 address = 2600:1408:1c::40
```

6.2 Dig

Es otro programa utilizado para preguntar a los servidores DNS.

Herramienta utilizada para solucionar problemas de DNS gracias a su flexibilidad, facilidad de uso y claridad en la presentación de la información. Normalmente, *dig* se usa pasándole argumentos desde la línea de comandos (CLI), pero también tiene un modo de operar por lotes, leyendo las consultas desde un archivo.

Este comando funciona tanto en sistemas operativos UNIX/Linux como en Windows.

```
sergio@Debian-DAW:~$ dig cisco.com a

; <<>> DiG 9.20.11-4-Debian <<>> cisco.com a
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50676
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;cisco.com.                     IN      A

;; ANSWER SECTION:
cisco.com.              1108    IN      A       72.163.4.185

;; Query time: 16 msec
;; SERVER: 80.58.61.250#53(80.58.61.250) (UDP)
;; WHEN: Sun Sep 14 11:24:46 CEST 2025
;; MSG SIZE  rcvd: 54
```

```
sergio@Debian-DAW:~$ dig cisco.com mx

; <<>> DiG 9.20.11-4-Debian <<>> cisco.com mx
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30670
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;cisco.com.                     IN      MX

;; ANSWER SECTION:
cisco.com.              1800    IN      MX      30 aer-mx-01.cisco.com.
cisco.com.              1800    IN      MX      10 alln-mx-01.cisco.com.
cisco.com.              1800    IN      MX      20 rcdn-mx-01.cisco.com.

;; Query time: 140 msec
;; SERVER: 80.58.61.250#53(80.58.61.250) (UDP)
;; WHEN: Sun Sep 14 11:25:13 CEST 2025
;; MSG SIZE  rcvd: 118
```