CAPÍTULO 3
TRADUCIENDO NOMBRES A DIRECCIONES (DNS)

ANDRES BARBIERI

En este capítulo se mostrarán más detalles sobre el sistema de DNS, actor necesario para que el requerimiento HTTP realizado por el usuario tenga éxito, obteniendo así la información contenida en la URL: `http://www.unlp.edu.ar`. El texto no entrará en los detalles de cuestiones como el formato específico del mensaje de DNS y todos los tipos de registros que define. Para esto se puede consultar la bibliografía [FS11], [KR12] y [LA06].

Introducción

El servicio de DNS (Domain Name System) tiene la particularidad que, en la mayoría de los casos, no es utilizado directamente por los usuarios o aplicaciones, sino que funciona como un apoyo al resto de los servicios de Internet. Podríamos decir que no existen user-agents para este sistema, aunque sí varias herramientas como pueden ser: `dig(1)`, `nslookup(1)` y `host(1)`. Su objetivo principal es el de traducir nombres de dominio a direcciones de Internet (direcciones IP) y, de esta forma, lograr una abstracción de las direcciones de red utilizadas internamente por los protocolos del stack TCP/IP. Esto permite ubicar a un dispositivo, o un nodo en la red, por su nombre sin importar cual es la dirección IP que tienen asignada en ese momento. Otra ventaja es que los usuarios personas, no necesitan recordar las direcciones IP. Para IPv4 (32 bits), recordar cuatro números de a lo sumo tres dígitos no parece tan complicado, el problema surge cuando se deben recordar muchos de estos valores y el caso parece empeorar con la necesidad de recordar direcciones IPv6 (128 bits).

El núcleo del DNS es un sistema distribuido de forma jerárquica, el cual está conformado por servidores ubicados a lo largo y ancho de todo el mundo. Cada servidor tendrá la responsabilidad de mantener parte de la información dentro de la jerarquía del espacio de nombres. Para mayor escalabilidad y tolerancia, los servidores, por lo general, tienen réplicas funcionando de forma activo-activo. Este sistema es apoyado por servidores intermedios o de cache, generalmente locales, a los cuales se suman los clientes como miembros.

Historia

El hecho de utilizar nombres en lugar de direcciones data de 1973, cuando todo se mantenía en un archivo global: `HOSTS.TXT`, que era soportado por el SRI (Stanford Research Institute, hoy SRI International). Dentro del SRI, esta tarea dio lugar a la creación del NIC (Network Information Center). El servicio funcionaba de forma completamente centralizada. El archivo `HOSTS.TXT` estaba disponible para "bajar" utilizando el protocolo FTP y las modificaciones al mismo eran solicitadas vía e-mail.

Cada determinado tiempo se juntaban las solicitudes de actualizaciones, se cambiaba el archivo y se generaba una nueva versión disponible al público. El sistema centralizado aseguraba que el NIC asignara las direcciones de forma consistente, pero tenía el problema de no ser escalable ni tolerante a fallas.

Para 1980, este método era muy difícil de mantener por lo que empezaron las investigaciones para reemplazarlo. El encargado de diseñar el nuevo sistema fue Paul Mockapetris, del USC (University of Southern California) Information Sciences Institute, y para fines de 1983, se generaron los documentos RFC-882[Moc83a] y RFC-883[Moc83b], que luego fueron reemplazados, en 1987, por RFC-1034[Moc87a] y RFC-1035[Moc87b]. El sistema diseñado por Mockapetris trabaja de forma distribuida, pero concentra el comienzo de las búsquedas en un conjunto de servidores raíces y luego delega de forma jerárquica. Existen varias RFC definidas dentro del marco del IETF que aportan más detalles y enriquecen el protocolo, como pueden ser sobre la seguridad con DNSSEC[AAL$^+$05a][AAL$^+$05b].

La primera implementación de un servidor de DNS fue realizada en 1984 en la Universidad de Berkeley y ejecutaba sobre Unix BSD. Más tarde, al ser re-escrita por otros estudiantes de la misma universidad, fue nombrada BIND (Berkeley Internet Name Domain). En la actualidad, hay diferentes versiones del servidor DNS mantenidas por el ISC (Internet Systems Consortium), que fue fundado por Paul Vixie en 1994. Además, ha sido portado a una gran cantidad de plataformas.

DNS en acción

El requerimiento de DNS

Desde el host *n11-client* el usuario ha "tipeado" en su navegador, o más precisamente aplicación de user-agent, la URL `http://www.unlp.edu.ar`. La aplicación deberá encargarse de encaminar el requerimiento HTTP hacia el servidor (podría estar implementado de distintas formas, como podría ser: por un servicio montado sobre un dispositivo físico, un sistema embebido, una máquina virtual, etc.) en la red que sirve ese recurso web. El servidor será alcanzado por su identificador único, la dirección IP mediante la cual se conecta a Internet. La aplicación que utiliza el usuario contactará al sistema de DNS para solicitarle la traducción del nombre de dominio: *www.unlp.edu.ar*, contenido dentro de la URL, a su identificador IP. La URL además del nombre de dominio, puede contener otras características como el protocolo, el recurso dentro del dominio, etc. Podría estar formada de la siguiente forma: `http://www.unlp.edu.ar:8080/index.html`.

El funcionamiento del DNS se realiza en un esquema cliente/servidor, siendo el cliente, en este caso la aplicación, que requiere la resolución de nombres. El código (instrucciones de máquina) del cliente de DNS se lo agrupa en un módulo de software llamado **Resolver** (en un intento de castellanización podríamos llamarlo "Resolvedor"). El **Resolver** se lo podría considerar como un agente encargado de solucionar las solicitudes del cliente utilizando la infraestructura de DNS. En este caso lo consideramos como parte del user-agent mismo, el que usa el usuario para solicitar el recurso (i.e. el navegador o browser). En el ejemplo la herramienta utilizada es `curl(1)`.

El **Resolver** deberá contactar al **Servidor de DNS local** asignado al host cliente para solicitarle esta traducción de nombre de dominio a dirección IP. El servidor de DNS local es contactado directamente por su dirección IP, que el host cliente obtuvo mediante su configuración, para el ejemplo la configuración se

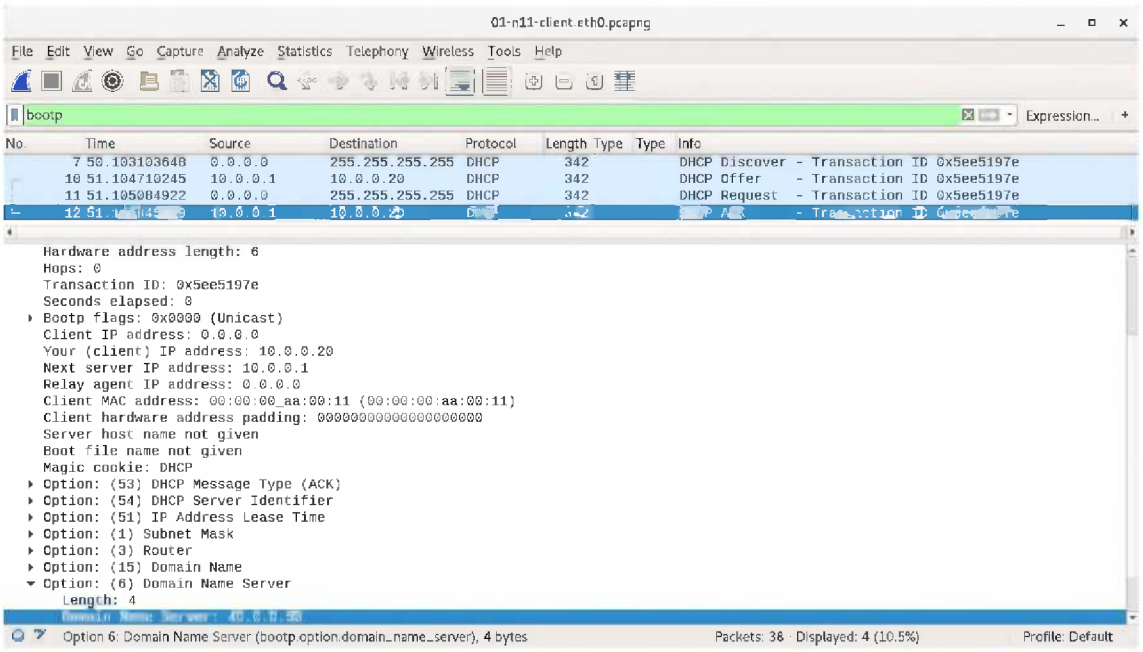

Figura 3.1: Captura de auto-configuración por DHCP del cliente, asignación de servidor de DNS.

realizó de forma automática mediante el protocolo DHCP[Dro97] (ver. figura 3.1 del archivo de captura `01-n11-client.eth0.pcapng`).

Usando el protocolo de DNS, el cliente solicitará a su DNS local y esperará la traducción. El formato de los mensajes y este diálogo se puede ver en la figura 3.2 del archivo de captura `01-n11-client.eth0.pcapng` tomada en la interfaz del dispositivo cliente. En la misma se observa que se utiliza el protocolo de transporte UDP (User Datagram Protocol) y el port usado por el servidor de DNS local es el 53. Se ve que se consulta por el FQDN (Full Qualified Domain Name), en castellano *Nombre Completo de Dominio*, *www.unlp.edu.ar* y se recibe como respuesta la *IP: 163.10.0.72*. Entre el mensaje número 19, requerimiento, y su respuesta el mensaje número 20, hay aún detalles no revelados, que el sistema local de DNS debió encargarse de solucionar. El diálogo aquí descripto corresponde al identificado en la figura 3.3 entre *n11-client* y *n18-dns-local-resolver*.

En la estructura del mensaje, sin entrar en los detalles, se puede observar que lleva un identificador el mensaje: campo *Transaction ID*. Este campo, en conjunto con los números de ports, sirve para asociar la respuesta con la consulta correspondiente.

Parece importante aclarar al lector que el código del **Resolver** ejecutando en el user-agent, generalmente no se implementa como un servicio activo, sino como un conjunto de rutinas encapsuladas (software) en una biblioteca de funciones que se link-editan conjuntamente con el código de la aplicación. En los sistemas Unix o basados en sus principios, en este caso GNU/Linux, el `resolver(3)(5)` es un conjunto de funciones incluidas en la biblioteca/librería de funciones estándares **C library (libc)**. Este tipo de resolver no realiza forma alguna de caching y se lo suele llamar **Stub/Dumb Resolver**. El encargado del caching, en estos casos, es el **Servidor de DNS local**, que a veces, también se lo suele llamar **Resolver**. Existen otras implementaciones que tienen un resolver activo, conocidos como **Smart Resolver**, funcionando en cada equipo como si fuese un **Servidor de DNS local**. Estos permiten realizar caching u ofrecer funcionalidades extras, como, por ejemplo, manejo de estadísticas o selección de los servidores con

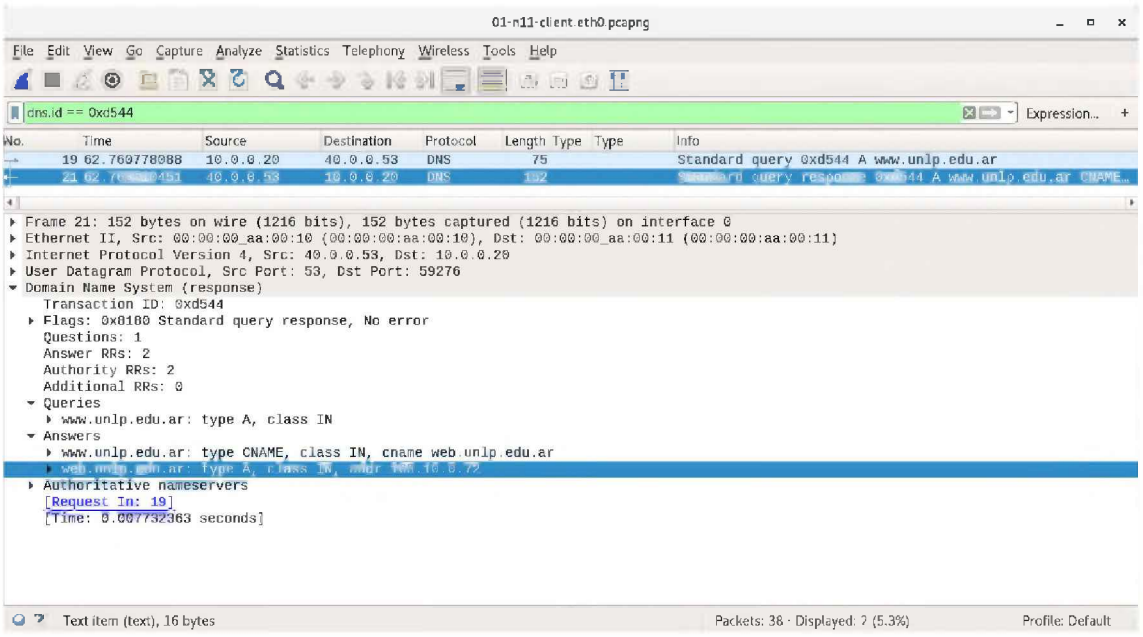

Figura 3.2: Captura del diálogo de DNS entre cliente y servidor local de DNS.

menor RTT. En la actualidad, también existen servidores de DNS públicos funcionado como **Servidor de DNS local**, que se encuentran distribuidos en la Internet, a menudo replicados, como es el caso de los **Public DNS**. Estos son servidores de DNS globales utilizables como locales. Mas detalles se podrán encontrar en URLs como: `https://en.wikipedia.org/wiki/Public_recursive_name_server`.

Espacio de nombres, FQDN

En párrafos anteriores se mencionó que habían detalles sin resolver. ¿Cómo obtiene el **Servidor de DNS local** la respuesta que entrega al cliente, donde indica que *www.unlp.edu.ar* mapea a la *IP: 163.10.0.72*? Para resolver esta cuestión el servidor deberá interrogar al núcleo del DNS, que, como se mencionó, es un sistema distribuido de forma jerárquica.

Para lograr esta distribución jerárquica se requiere que el espacio de nombres utilizados también así lo sea. Si se presta atención, los FQDN (ver figura 3.4) están formados por una lista de labels (etiquetas) separadas por puntos que van desde el nodo, la etiqueta más a la izquierda, hasta la raíz del árbol, el punto. De esta forma se puede pensar que la etiqueta de la derecha está enganchada en la raíz y a medida que vamos pasando por los puntos se va "subiendo" a la copa del árbol, o de otro modo, al tope de la jerarquía. Gráficamente, el árbol estaría invertido. Ver de ejemplo la figura 3.5.

Otras características que se pueden mencionar de los FQDN es que son "case-insensitive" (es decir, mayúsculas y minúsculas dan igual) y cada etiqueta puede tener como máximo 63 caracteres. La cantidad máxima de etiquetas es 127 y el nombre no puede tener más de 255 caracteres en total. Se distingue por terminar con un punto, ".", (implícito si no lo tuviese) y se le puede agregar/concatenar, al momento de hacer la búsqueda, un espacio de dominio/sub-dominio. Si el nombre sólo tiene una etiqueta se le agrega un dominio, si tiene más eslabones se lo considera como un FQDN y solo se le agrega el punto final de forma implícita al tratarlo. Existe, también, la posibilidad de escribir nombres de dominios con juegos de símbolos no ASCII llamados Internationalized Domain Name for Applications

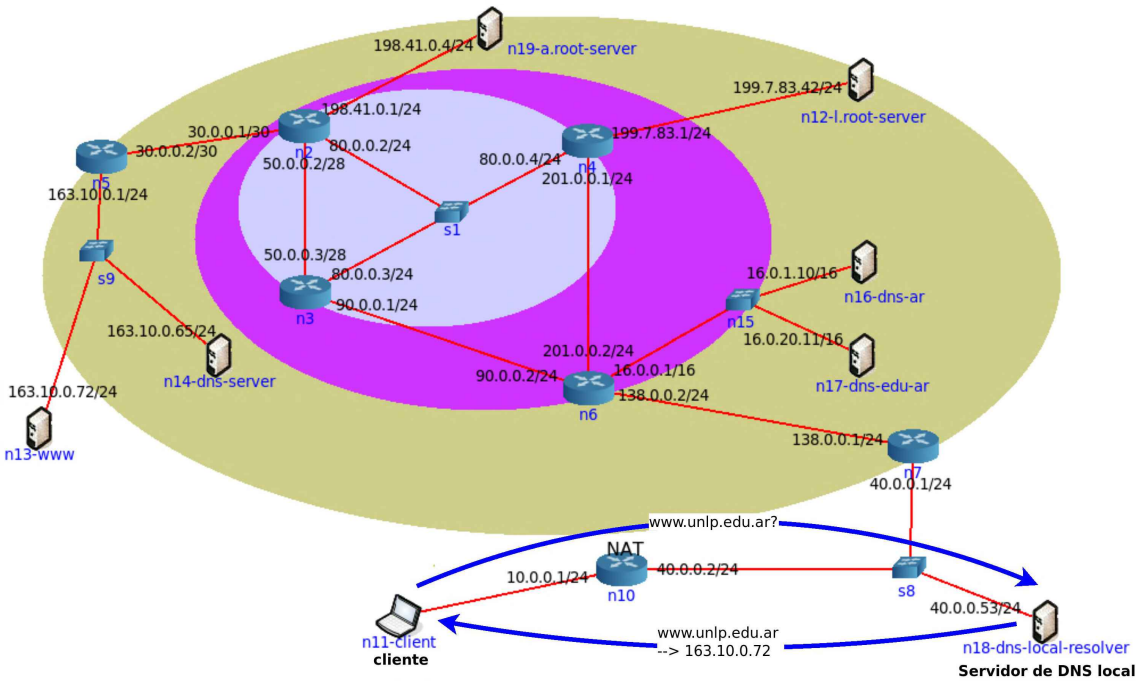

Figura 3.3: Diálogo dentro del sistema DNS entre cliente y servidor local.

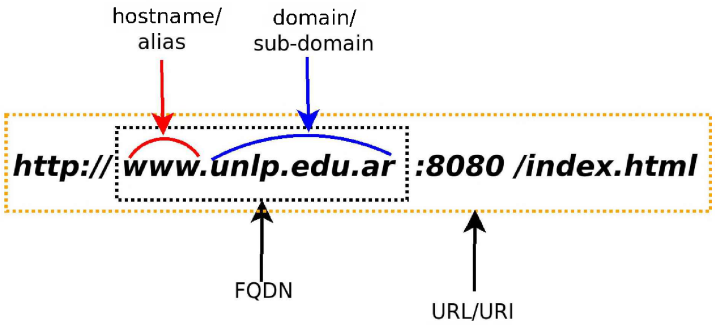

Figura 3.4: URL y FQDN (Nombre completo de dominio).

(IDNA) [Dro10]. La siguiente lista son ejemplos de nombres de dominios, el primero Non-FQDN y el resto FQDN.

- *www* (No FQDN)
- *www.unlp.edu.ar.* (FQDN)
- *www.unlp.edu.ar* (considerado FQDN)
- *web.ru.*
- *www.net.*
- *ada.info.unlp.edu.ar.*
- *www.l.google.com.*
- *other.com.*
- *30.8.10.163.in-addr.arpa.*

Dominios y sub-dominios

Para el ejemplo de *www.unlp.edu.ar* se considera que *unlp.edu.ar* es el dominio para el Non-FQDN *www*. Para el caso del dominio *unlp.edu.ar*, *unlp.edu.ar* es un sub-dominio dentro del dominio *edu.ar* y así lo será *edu.ar* dentro de *ar*. Otros ejemplos serían: *info.unlp.edu.ar* y *psico.unlp.edu.ar* son sub-dominios dentro de *unlp.edu.ar*. Un concepto similar a dominio, que a veces resulta un tanto confuso, es el de zona. Este término esta más relacionado con el despliegue de los servidores y lo veremos más adelante.

TLD (Top Level Domain)

Los dominios de primer nivel en la jerarquía, los que se encuentran enganchados a la raíz, son conocidos como TLD (Top Level Domains), y están predefinidos, aunque pueden aparecer nuevos. Actualmente existen TLD genéricos conocidos como gTLD que contienen dominios con propósitos particulares, de acuerdo a diferentes actividades. Para 1980, los gTLD eran: **.com, .edu, .gov, .int, .mil, .net** y **.org**, pero sólo **.com, .net** y **.org** tenían abierto el registro, el resto estaban dedicados (Sponsored TLD). Entre 2011 y 2012 el ICANN abrió los gTLD permitiendo registrar nuevos nombre de dominio en el tope de la jerarquía. El programa indica un proceso de licitación y remate. Para 2015 había más de 500 nuevos dominios, como **.beer, .paris**.

De forma paralela a los gTLD existen los ccTLD (Country-Code TLD) que contienen dominios delegados a los diferentes países del mundo. Los códigos de los paises están codificados en dos símbolos, habitualmente letras, y actualmente se admiten símbolos o letras de otros alfabetos. Se encuentran, por ejemplo, en este formato **.ar (Argentina), .tv (Tuvalu), .zw (Zimbabwe), .uk (Reino Unido), .ru (Rusia), .us (Estados Unidos), etc**.

También existen otros dominios con propósitos específicos como el **.ARPA.TLD** que es un dominio especial, llamado de infraestructura, usado internamente por los protocolos para resolución de reversos: de direcciones IP a nombres.

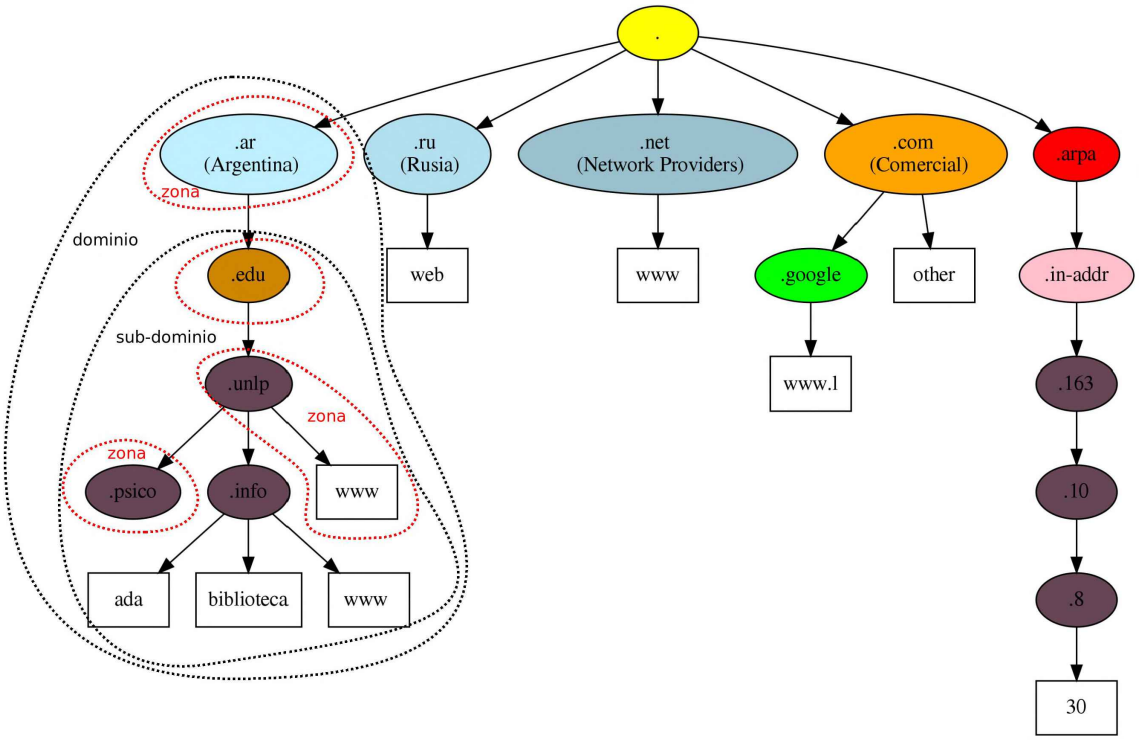

Figura 3.5: Ejemplos de nombres de dominio.

Jerarquía de DNS, servidores de dominios

Una vez organizado el espacio de nombres de forma jerárquica mediante dominios y sub-dominios se lo distribuye en los diferentes servidores que implementan la infraestructura de Internet. Esta asignación también conlleva delegar la administración.

Servidores raíces (Root name servers)

Comenzando desde la raíz, **"."**, es necesario asignar los servidores raíces, es decir, los servidores que dan origen a las delegaciones o distribución del sistema. Son los que conocen a que servidores delegar la responsabilidad de cada TLD. En la actualidad, existen 13 **Root Name Servers** distribuidos en todo el mundo, de los cuales varios no son servidores únicos, sino que trabajan con redundancia y las réplicas están distribuidos geográficamente. Con esta redundancia, combinada con *Anycast*, se logra que los requerimientos se atiendan según la proximidad del punto de vista de la red, logrando mayor eficiencia (i.e. mejores tiempos de respuesta). La cantidad de servidores raíces está acotada a 13 debido a una limitación inicial de 512 bytes de un mensaje de DNS sobre UDP. A continuación se listan los servidores raíces disponibles en el año 2015.

```
.   518400  IN      NS      A.ROOT-SERVERS.NET. # Versign-grs.com
.   518400  IN      NS      B.ROOT-SERVERS.NET. # ISI.edu
.   518400  IN      NS      C.ROOT-SERVERS.NET. # Cogent.com (ANYCAST)
.   518400  IN      NS      D.ROOT-SERVERS.NET. # UMD.edu (Univ. Maryland)
.   518400  IN      NS      E.ROOT-SERVERS.NET. # NASA.gov
.   518400  IN      NS      F.ROOT-SERVERS.NET. # ISC.org (ANYCAST)
.   518400  IN      NS      G.ROOT-SERVERS.NET. # NIC.mil
.   518400  IN      NS      H.ROOT-SERVERS.NET. # ARMY.mil
.   518400  IN      NS      I.ROOT-SERVERS.NET. # NIC.ddn.mil (ANYCAST)
.   518400  IN      NS      J.ROOT-SERVERS.NET. # Versign-grs.com (ANYCAST)
.   518400  IN      NS      K.ROOT-SERVERS.NET. # RIPE.net (ANYCAST)
.   518400  IN      NS      L.ROOT-SERVERS.NET. # ICANN.org (ANYCAST)
.   518400  IN      NS      M.ROOT-SERVERS.NET. # WIDE.ad.jp (ANYCAST)

A.ROOT-SERVERS.NET.     3600000 IN      A       198.41.0.4
```

Figura 3.6: Distribución de los ROOT Servers en 2015 (Fuente: *www.root-servers.org.*)

```
A.ROOT-SERVERS.NET.      3600000 IN      AAAA    2001:503:ba3e::2:30
B.ROOT-SERVERS.NET.      3600000 IN      A       192.228.79.201
C.ROOT-SERVERS.NET.      3600000 IN      A       192.33.4.12
D.ROOT-SERVERS.NET.      3600000 IN      A       128.8.10.90
E.ROOT-SERVERS.NET.      3600000 IN      A       192.203.230.10
F.ROOT-SERVERS.NET.      3600000 IN      A       192.5.5.241
F.ROOT-SERVERS.NET.      3600000 IN      AAAA    2001:500:2f::f
G.ROOT-SERVERS.NET.      3600000 IN      A       192.112.36.4
H.ROOT-SERVERS.NET.      3600000 IN      A       128.63.2.53
H.ROOT-SERVERS.NET.      3600000 IN      AAAA    2001:500:1::803f:235
I.ROOT-SERVERS.NET.      3600000 IN      A       192.36.148.17
J.ROOT-SERVERS.NET.      3600000 IN      A       192.58.128.30
J.ROOT-SERVERS.NET.      3600000 IN      AAAA    2001:503:c27::2:30
```

Esta lista se puede obtener vía una consulta de DNS o usando FTP[PR85].

En la figura 3.6 se muestra la distribución de los **Root Name Servers** en el mundo para 2015. En la actualidad, el gráfico se torna bastante más cargado de hitos en el mapa.

Cada servidor de DNS debe conocer al menos uno de esos servidores y poder alcanzarlo para comenzar a operar. En la topología de ejemplo se han definido solo 2 root servers: *n19-a.root-server* y *n12-l.root-server*. El **Servidor de DNS local** del ejemplo, *n18-dns-local-resolver*, debe contactarlos para comenzar a operar.

Servidores TLD y de niveles inferiores

La distribución del servicio se logra mediante la delegación jerárquica. Como se indicó, los servidores raíces deben conocer a los servidores responsables de cada TLD. Esto se conoce como delegación de zona de DNS. La zona sería el dominio sin los sub-dominios delegados, gráficamente, un eslabón en la cadena de delegación. La delegación consiste en saber cuales son los servidores que se encargan de resolver (o sub-delegar) los sub-dominios, llamados en la implementación del servicio como zonas. La delegación se realiza de manera **autoritativa**. Un servidor **autoritativo** tiene toda la información para una zona, puede producir cambios sobre la misma y es el que tiene la última versión. Puede existir más de un servidor autoritativo para una zona y es lo recomendable para ofrecer tolerancia a fallas y

probablemente distribución de carga. De este modo, para el ccTLD **.ar**, se podría obtener la siguiente delegación:

```
ar.                     7172        IN        NS        c.dns.ar.
ar.                     7172        IN        NS        a.dns.ar.
ar.                     7172        IN        NS        e.dns.ar.
ar.                     7172        IN        NS        f.dns.ar.
ar.                     7172        IN        NS        ar.cctld.authdns.ripe.net.
ar.                     7172        IN        NS        b.dns.ar.
ar.                     7172        IN        NS        d.dns.ar.
c.dns.ar.               86400        IN        A        200.108.148.50
...
```

Esto mismo sucederá con otros TLDs. Para que un TLD pueda operar debe ser cargada la delegación en los servidores raíces. Luego, de forma similar sucederá que los servidores encargados de los TLD, como el ccTLD **.ar**, tendrán que saber delegar los sub-dominios o zonas, por ejemplo, **.edu.ar**, **.com.ar**, **.org.ar**, etc. La forma de organizar cada dominio es potestad exclusiva de a quien se lo delega. En particular, en la Argentina se ha tomado una política similar que en los TLD y se han generado sub-dominios con etiquetas similares y estos se han sub-delegado generando un estructura similar a la de "arriba". Podría suceder que el dominio de nivel superior no delegue en zonas e implemente los niveles de etiquetas dentro de la misma. Bajando en la jerarquía irá sucediendo lo mismo con respecto a las delegaciones y a la creación de zonas.

En la topología utilizada se implementa de forma simplificada solo un TLD: **.ar** y se sirve con un solo nodo: *n16-dns-ar*(*a.dns.ar*) Bajo este se implementa el dominio **edu.ar** hosteado por los nodos *n17-dns-edu-ar*(ns1.riu.edu.ar) y *n14-dns-server*(*anubis.unlp.edu.ar*). El sub-dominio:
unlp.edu.ar está implementado en el nodo servidor *n14-dns-server* que es *anubis.unlp.edu.ar*. Las direcciones IP correspondientes a los nodos *a.dns.ar*, *b.dns.ar* y *ns1.riu.edu.ar* son distintas a las reales. Para *a.dns.ar* se usa 16.0.1.10 y para *ns1.riu.edu.ar* 16.0.20.11. En las figuras 3.7 se muestra como está desplegado en la topología.

La resolución de la consulta

Volvamos al **Servidor de DNS local** recibiendo la consulta del FQDN: *www.unlp.edu.ar* desde el cliente. Ahora, el servidor tiene que arreglárselas para responder y va a comenzar consultando desde la raíz del sistema. En este caso va a preguntar a algunos de los **Root Name Servers** que conoce.

En la figura 3.8 del archivo de captura `03-n18-dns-local.eth0.pcapng` tomada en la interfaz del dispositivo del servidor local de DNS, *n18-dns-local-resolver*, se puede ver la respuesta del servidor raíz consultado por el DNS local. La respuesta recibida no es la traducción solicitada sino los registros de delegación, una "pista" para que el servidor local se aproxime a la respuesta. Si se continúa analizando las consultas y las respuestas que va recibiendo se puede observar que el comportamiento es el mismo, no recibe la traducción, sino los registros de delegación. Solo cuando llega al servidor autoritativo para el dominio *unlp.edu.ar* recibe la respuesta esperada.

En la figura 3.9 se muestra el diagrama de flujos/interacción entre el servidor local y los servidores que forman parte de la jerarquía de DNS en la topología de ejemplo.

1. Primero consulta a un servidor raíz, *a.root-server.net* (198.41.0.4).
2. Luego consulta a un servidor del TLD extbf.ar *a.dns.ar* (16.0.1.10).
3. Finalmente a *anubis.unlp.edu.ar* (163.10.0.65).

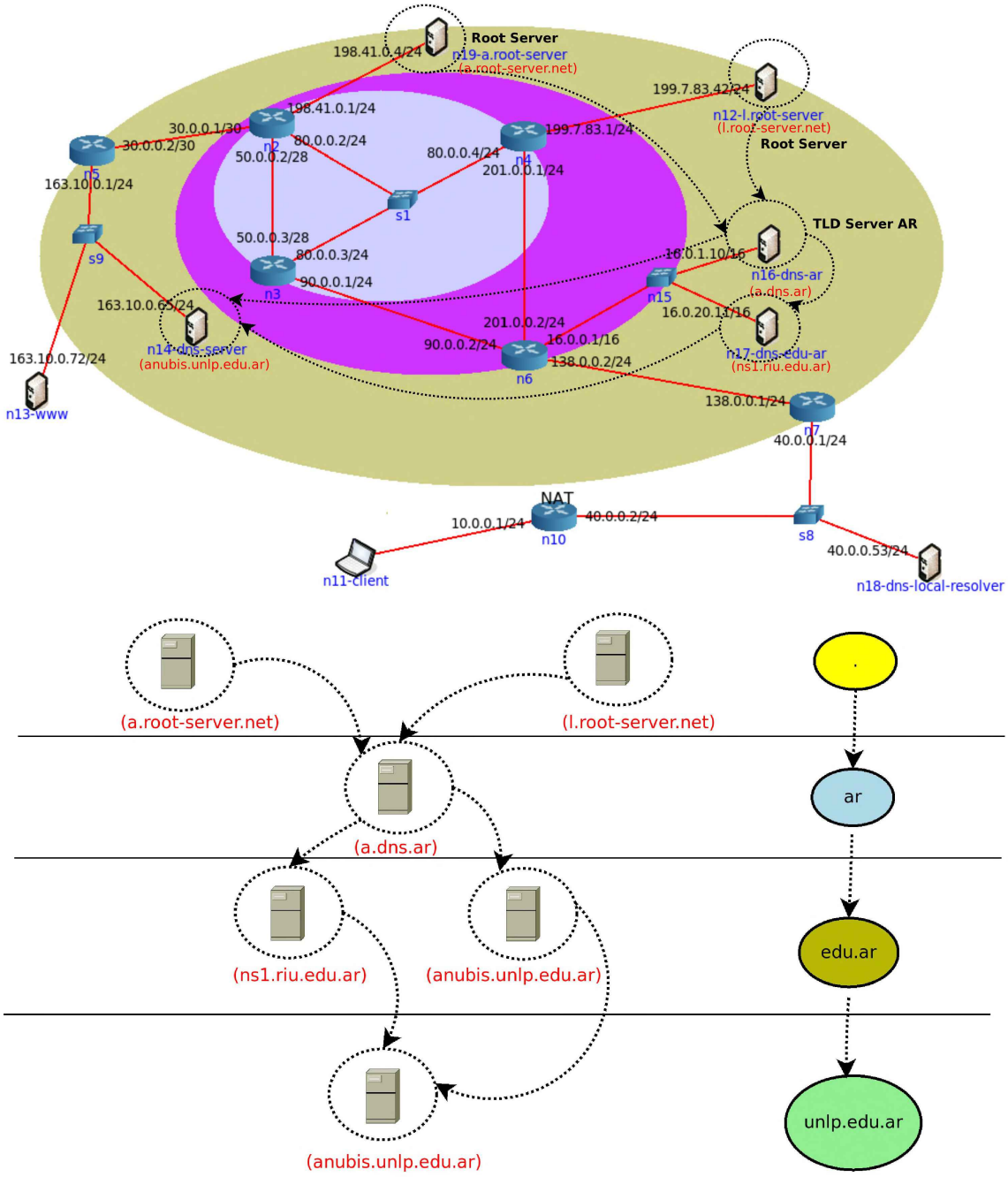

Figura 3.7: Delegación de los sub-dominios en la topología de test.

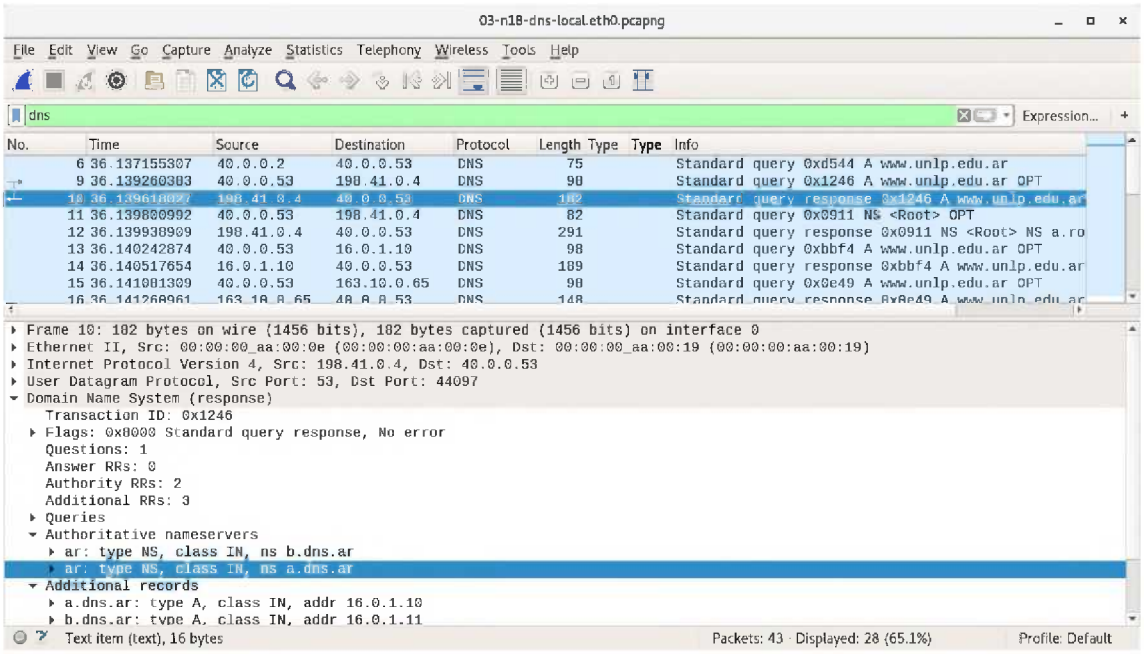

Figura 3.8: Captura del diálogo de DNS entre local y los servidores autoritativos.

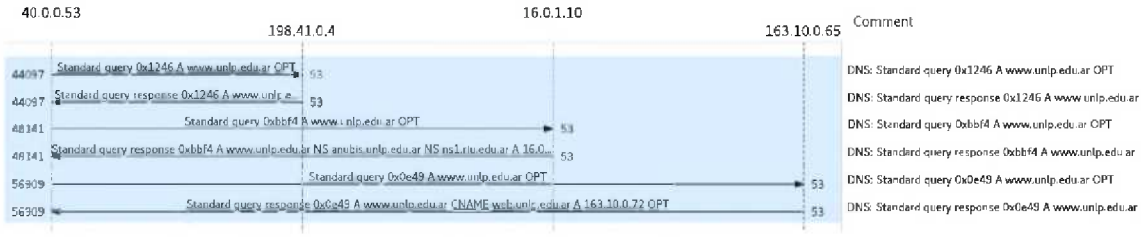

Figura 3.9: Diagrama de interacción entre el DNS local y los servidores autoritativos.

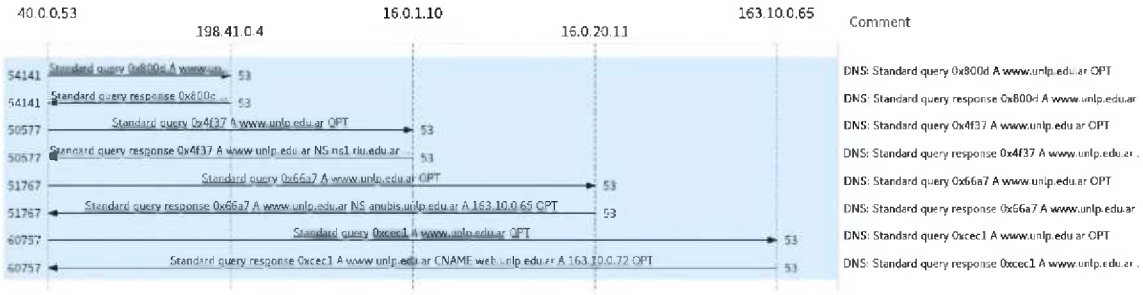

Figura 3.10: Diagrama alternativo de interacción entre el DNS local y los servidores autoritativos.

Pareciese que saltease el dominio **.edu.ar** y consulta directamente a uno de los servidores del dominio **unlp.edu.ar**. Esto no es así, ya que el servidor *anubis.unlp.edu.ar* funciona para ambos dominios y en este caso puede responder de forma autoritativa directamente con la traducción deseada. Otro posible escenario podría ser el mostrado en la figura 3.10 del archivo de captura `03-n18-dns-local.eth0-v2.pcapng`

1. Consulta a un servidor raíz, *a.root-server.net* o *l.root-server.net*.

2. Consulta a un servidor del TLD **.ar** *a.dns.ar* (16.0.1.10).

3. Luego consulta a un servidor del sub-dominio **edu.ar**, donde podría ser *ns1.riu.edu.ar* (16.0.20.11).

4. Finalmente a *anubis.unlp.edu.ar* (163.10.0.65) de quien recibe la traducción.

Con una secuencia o la otra el servidor local ha obtenido la respuesta donde se le indica que *www.unlp.edu.ar* es un alias a *web.unlp.edu.ar* mediante un registro de tipo **"CNAME"** y que este tiene la *IP: 163.10.0.72* según el registro **"A"** devuelto.

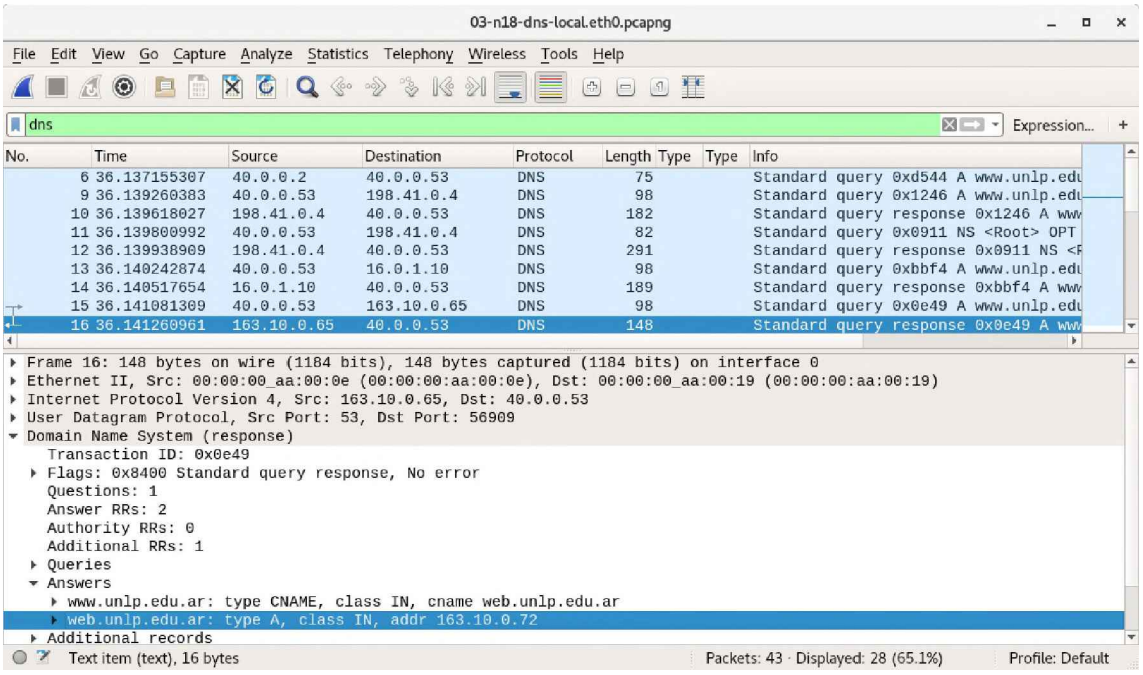

Figura 3.11: Captura del diálogo de DNS entre local y el autoritativo de unlp.edu.ar.

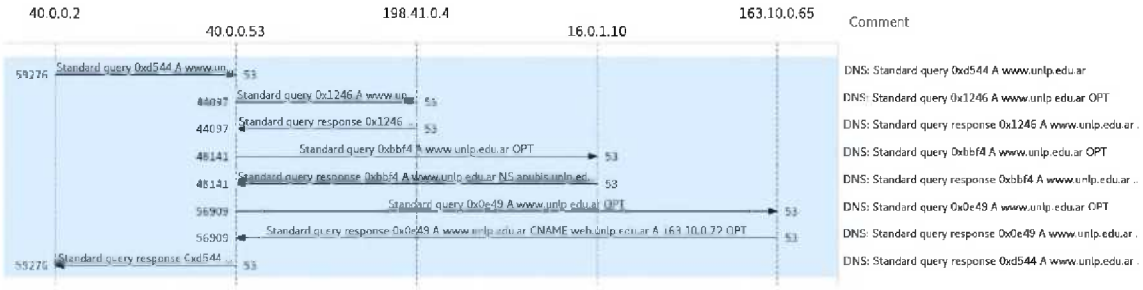

Figura 3.12: Diagrama de interacción entre el cliente, el DNS local y los autoritativos.

El requerimiento y la respuesta completa

Ahora, el **Servidor de DNS local** ha obtenido la respuesta. En la misma tiene la información solicitada por el cliente al cual le deberá entregar la información. Es la respuesta que se muestra en la figura 3.2. El diagrama de interacción completo se puede ver en la figura 3.12.

Al recibir la respuesta de su **Servidor de DNS local** el cliente ya tiene la información necesaria para construir el requerimiento HTTP y contactar al servidor web del sito `http://www.unlp.edu.ar`, como se podrá aprender en el capítulo correspondiente.

Una cuestión que puede resultar extraña al lector, si compara la captura tomada en la interfaz del cliente con la tomada en la interfaz del servidor local, es que el requerimiento de DNS sale con la dirección IP origen 10.0.0.20 y llega al destino 40.0.0.53 con la dirección origen 40.0.0.2. Eso se debe a un proceso de traslación de direcciones llevada a cabo por el router que da servicio al cliente. Este mecanismo se llama NAT, o más precisamente NAPT [SH99], y no es recomendado por romper el principio de end-to-end[Car96]. Hoy en día es un requerimiento de facto debido a la escasez de direcciones IPv4.

Otros Detalles

Consultas recursivas e iterativas

Si se analiza el escenario y los actores se pueden observar dos formas de consultar o de resolver la petición. Una es la que utiliza el cliente con su servidor, en la cual al hacer la petición se desentiende de los detalles y espera la respuesta final. Este método se llama *Recursivo* y se puede divisar en los flags de los mensajes de DNS. La otra forma es la que utiliza el servidor local. En el segundo caso, él mismo tiene que hacer el trabajo de ir consultando paso por paso la jerarquía de DNS hasta obtener la respuesta que entregará al cliente. Este método se llama *Iterativo* y también se puede divisar en los flags. Esta comparación se puede hacer observando las figuras 3.13. En la consulta recursiva se espera que el servidor al cual se indaga resuelva la cuestión haciendo las búsquedas necesarias, o si tenemos suerte se entregará desde cache. Para el caso de las iterativas solo se espera que el servidor responda con lo que el conoce, en particular la información sobre la cual tiene autoridad. Los servidores de DNS, en la mayoría de los casos restringen las consultas recursivas a los posibles clientes finales que atenderán. En este caso, el servidor local del ISP del ejemplo no atenderá consultas recursivas de clientes que no pertenezcan a su porción de red.

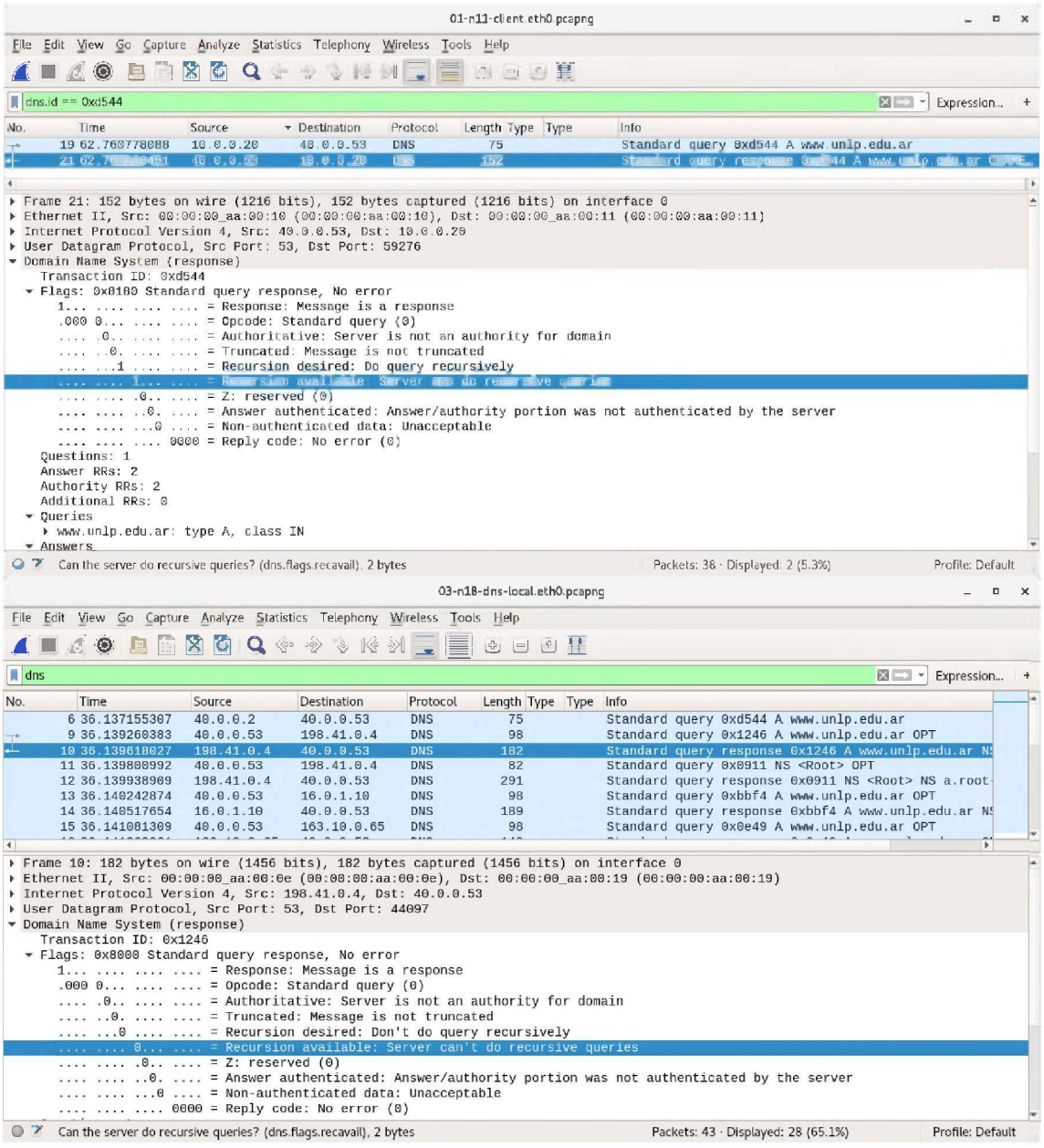

Figura 3.13: Flags que distinguen método recursivo de iterativo.

En el modelo de prueba también se tiene como "usuario" del servicio al servidor web, *n13-www*, o directamente *web.unlp.edu.ar*. En este caso el host que cumple las funciones de servidor http usará su propio servidor DNS local, *n14-dns-server* (*anubis.unlp.edu.ar*). Este es un caso particular donde el servidor de DNS funciona como autoritativo y DNS o resolver local.

Respuestas autoritativas

Otra cuestión importante con respecto a los servidores autoritativos y las respuestas autoritativas es identificarlas. Se aclaró que los servidores autoritativos son los responsables para un dominio o zona. Estos son los encargados de mantener la información de la misma y actualizarla cuando sea necesaria. Son los que se indican con los registros **"NS"**. Las respuestas que estos brinden de los registros que poseen irán marcadas con el flag de autoritativa. Los servidores que cacheen esta información, como es el caso de el **Servidor de DNS local**, mantendrán en cache esta información mientras el tiempo de cache señalado por el autoritativo sea válido. Ante las reiteradas consultas por los mismos registros podrán entregar esta información desde la cache. De esta forma, se ahorra hacer la búsqueda por cada consulta recibida y se entregará la información almacenada con un flag que indique que dicha respuesta no es autoritativa. Para ver el comportamiento autoritativo y no autoritativo se puede ver la captura de la respuesta que entrega *anubis* al servidor local de la red del ISP, `03-n18-dns-local.eth0.pcapng`, fila 16, y la que entrega el servidor local al cliente, `01-n11-client.eth0.pcapng`, fila 19, donde se marca que no es autoritativa. Si se volviese a consultar por la misma información al servidor local, este la entregaría desde la cache hasta que venza su temporizador de almacenamiento.

Mecanismos de sincronización ente servidores

Para un mejor funcionamiento del sistema se recomienda tener 2 o más servidores autoritativos para una misma zona. Esto ofrece tolerancia a fallas y un mejor funcionamiento permitiendo balance de carga entre los mismos. Para una configuración más flexible se selecciona a uno de los servidores autoritativos como el principal(master), o primario, y el resto se los llama secundarios o slaves. El protocolo de DNS propone un mecanismo de sincronización entre primarios y secundarios llamado **Transferencia de Zona (XFR)**. Este método como transporta más información que la respuesta a una consulta típica utiliza como protocolo de transporte TCP (Transport Control Protocol). El protocolo TCP debe estar siempre disponible en los servidores y, de acuerdo a la cantidad de información a transmitir cliente-servidor pueden optar cuál utilizar.

REFERENCIAS

[AAL+05a] R. Arends, R. Austein, M. Larson, D. Massey y S. Rose. «Rfc 4033: Dns security introduction and requirements», 2005.

[AAL+05b] R. Arends, R. Austein, M. Larson, D. Massey y S. Rose. «Rfc 4035: Protocol modifications for the dns security extensions», 2005.

[Car96] B. Carpenter(Ed.). «Rfc 1958: Architectural principles of the internet», 1996.

[Dro97] R. Droms. «Rfc 2131: Dynamic host configuration protocol», 1997.

[Dro10] R. Droms. «Rfc 5890: Internationalized domain names for applications (idna): Definitions and document framework», 2010.

[FS11] K. Fall y R. Stevens. «Tcp/ip illustrated, volume 1: The protocols, 2nd. ed.». Addison-Wesley, 2011. ISBN: 9780321336316.

[KR12] J. Kurose y K. Ross. «Computer networking: A top-down approach, 6th. ed.». Addison-Wesley, 2012. ISBN: 0132856204.

[LA06] Cricket Liu y Paul Albitz. «Dns and bind (5th edition)». O'Reilly Media, Inc., 2006. ISBN: 0596100574.

[Moc83a] P. V. Mockapetris. «Rfc 882: Domain names: Concepts and facilities», 1983.

[Moc83b] P. V. Mockapetris. «Rfc 883: Domain names: Implementation specification», 1983.

[Moc87a] P. V. Mockapetris. «Rfc 1034: Domain names: Concepts and facilities», 1987.

[Moc87b] P. V. Mockapetris. «Rfc 1035: Domain names: Implementation specification», 1987.

[PR85] J. Postel y J. Reynols. «Rfc 959: File transfer protocol (ftp)», 1985.

[SH99] P. Srisuresh y M. Holdrege. «Rfc 2663: Ip network address translator (nat) terminology and considerations», 1999.