

Introducció al sistema DNS

(Solució)

Arquitectura i Protocols d'Internet

Grau en Enginyeria Telemàtica

Grau en Enginyeria de Sistemes de Telecomunicació

**Frederic Raspall
Anna Agustí**

Departament d'Enginyeria Telemàtica
Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
Universitat Politècnica de Catalunya

CONTINGUTS

INTRODUCCIÓ AL SISTEMA DNS.....	3
1 Introducció	3
1.1 Motivació i perspectiva històrica	3
1.2 Què és el sistema DNS?	3
2 Conceptes bàsics i funcionament del sistema DNS.....	4
2.1 Espai de noms del DNS.....	4
2.2 Resource Records (RRs).....	4
2.3 Dominis i sub-dominis.....	5
2.4 Avantatges de l'espai de noms del DNS.....	5
2.5 Delegació de dominis	6
2.6 Servidors de noms i Fitxers de zona.....	6
2.7 El funcionament del DNS amb un exemple	7
2.8 Tipus de consultes DNS. RRs tipus NS.....	8
2.9 Caching i TTLs	9
2.10 Resolució de Noms. RRs tipus PTR.....	10
2.11 Àlies i noms canònics. RRs tipus CNAME.....	11
2.12 Tipus de Resource Records	11
2.13 Encaminament de correu electrònic. RRs tipus MX	12
2.14 Format textual dels RRs	12
2.15 Format dels noms de domini	13
2.16 Tipus de servidors de noms.....	13
2.17 RRs tipus SOA	14
2.18 Algunes funcionalitats més recents del DNS.....	14
2.19 DNS a xarxes IPv6. RRs tipus AAAA	15
2.20 El protocol DNS	16
2.21 Format dels fitxers de zona	17
2.22 Configuració de servidors DNS	18
2.23 Eines de diagnòstic del DNS	18
2.24 El DNS a Internet	20
2.25 Root Servers.....	21
2.26 Delegació de subdominis a in-addr.arpa. sense classes.....	22
ACTIVITATS AL LABORATORI	24
Objectius de la pràctica.....	24
Eescenari P08-E01	24
Exercici 1. Anàlisi del protocol DNS.....	25
Exercici 2. Configuració d'un servidor caché (caching-only nameserver).....	41
Exercici 3. Configuració d'un servidor mestre d'un domini.....	49
Exercici 4. Configuració d'un servidor esclau. Transferències de zona	57
Exercici 5. Delegació d'un subdomini	66
FIGURES.....	72

INTRODUCCIÓ AL SISTEMA DNS

1 INTRODUCCIÓ

1.1 MOTIVACIÓ I PERSPECTIVA HISTÒRICA

Les adreces IPv4 consten de 32 bits que s'acostumen a escriure en *dotted-quad* (a.b.c.d). Tot i que aquesta notació en simplifica la lectura, és evident que recordem millor paraules (amb un cert contingut semàntic, significat) que no pas números. Per exemple, és més senzill recordar <http://www.google.es> que <http://209.85.227.147>. Per aquest motiu, ja fa molts anys que els administradors de xarxa assignen noms a dispositius (sovint amb un cert context). Així, és comú anomenar màquines amb un nom que recordi la seva funcionalitat (p.ex. *ftp*, *www*, *printer*, etc.) o utilitzar noms d'entitats mitològiques (*atenea*, *zeus*, *apolo*, *minotaure*), de músics (*shubert*, *mozart*, *chopin*), de científics de renom (*euler*, *gauss*), de pintors (*renoir*, *gauguin*, *dali*), etc. El problema és que aquests noms no tenen res a veure amb l'adreçament del protocol IP. Cal, per tant, un mecanisme de traducció d'aquests noms (intel·ligibles per les persones) en identificadors intel·ligibles pel protocol IP.

En el context d'Internet, als anys 70, la Internet embrionària (l'anomenada ARPAnet) era molt petita i consistia en només uns centenars de hosts. El problema se solucionava fent servir un únic fitxer (anomenat HOSTS.TXT) que mantenía la correspondència entre noms i adreces i era gestionat pel *Network Information Center* (NIC) del *Stanford Research Institute* (SRI). Els administradors de cada xarxa enviaven canvis o noves parelles nom-adreça per e-mail al NIC, que actualitzava el fitxer un o dos cops per setmana; i els diferents administradors es descarregaven versions actualitzades d'aquest fitxer periòdicament. Quan ARPAnet va adoptar TCP/IP, el nombre de hosts va créixer molt i aquest sistema va deixar de ser viable degut a:

- L'elevat volum de tràfic i càrrega al NIC.
- Les col·lisions en els noms: era possible que dos administradors volguessin fer servir el mateix nom popular per una màquina, com *www* o *elvis*.
- A la poca flexibilitat i consistència en les dades degut al retard en actualitzar el fitxer.

Aquests problemes es devien fonamentalment a que el sistema basat en un fitxer era **centralitzat** i a que l'espai de noms era "**pla**". Per resoldre el problema, els responsables d'ARPAnet van endegar projectes de recerca per a desenvolupar un sistema sense aquestes limitacions. Finalment, el responsable del disseny de la nova arquitectura va ser en Paul Mockapetris qui, l'any 1984, va publicar els RFCs 882 i 883. Així va néixer el **Domain Name System (DNS)**. L'any 1987 es van acceptar com a estàndard els **RFCs 1034 i 1035**, que reemplacen els anteriors i constitueixen l'especificació actual del DNS. D'aleshores ençà s'han publicat molts RFCs que descriuen problemes de seguretat o d'implementació, o que afegeixen noves funcionalitats al sistema com autenticació, actualitzacions dinàmiques, etc.

1.2 QUÈ ÉS EL SISTEMA DNS?

El DNS es podria definir com una base de dades **distribuïda** amb informació sobre **hosts i serveis**. Això permet un **control local** dels "segments" de la base de dades i, a la vegada, permet que cada segment sigui **accessible a tota la xarxa** a través d'un esquema **client-servidor**. El sistema assoleix robustesa i rendiment mitjançant **replicació i caching**.

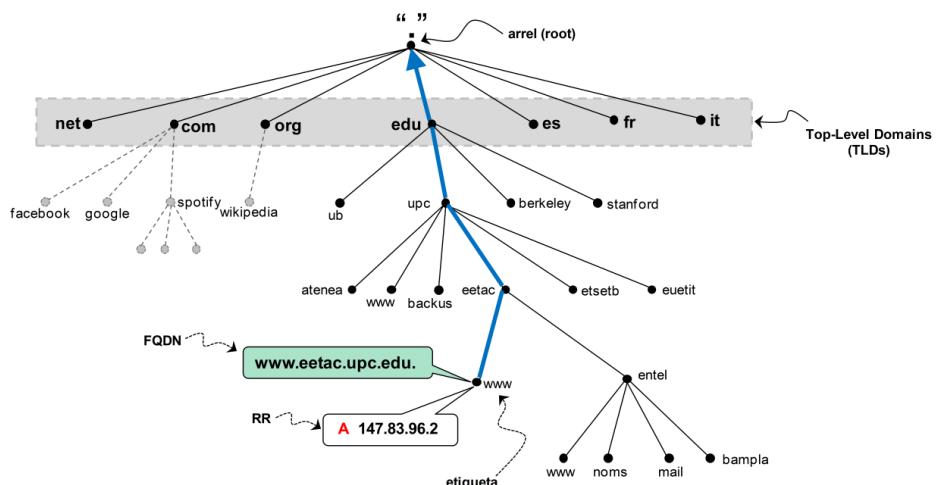
L'**arquitectura** del **DNS** inclou els següents elements o entitats:

- Un **espai de noms** de domini que permet anomenar hosts i indexar les dades a la base de dades.
- Els anomenats **Resource Records** (RRs): contenen la informació en sí de la base de dades.
- **Servidors de noms** (*nameservers*): són servidors d'informació de la base de dades, és a dir de RRs.
- **resolvers**: són entitats demandants d'informació, clients. Constitueixen la intereficie amb les aplicacions (per exemple, un navegador o un client de correu), que demanen al **resolver** resoldre un nom o adreça.
- Un protocol per sol·licitar/respondre dades en la base de dades del tipus petició-resposta. Els **resolvers** són sempre clients, mentre que els **nameservers** poden ser clients i servidors a la vegada.

2 CONCEPTES BÀSICS I FUNCIONAMENT DEL SISTEMA DNS

2.1 ESPAI DE NOMS DEL DNS

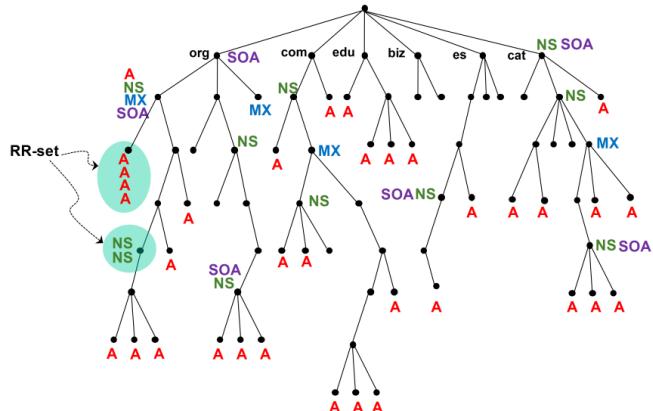
El DNS utilitzà un espai de noms jeràrquic, en forma d'arbre, amb un node arrel (*root*). En aquest arbre, cada node té una etiqueta. L'etiqueta associada a l'arrel és un punt. De la mateixa manera que en un sistema Unix/Linux, els diferents fitxers o directoris poden tenir noms absoluts (com ara `/home/API/dns.pdf`) o relatius (com `dns.pdf`, relatiu a `/home/API`), en l'espai de noms del DNS, el **nom de domini** d'un node de l'arbre pot ser **relatiu** (a un cert node) o **absolut**; en terminologia DNS, un **Fully-Qualified Domain Name (FQDN)**. Al DNS, el nom absolut o FQDN associat a un node és la concatenació d'etiquetes des del node fins a l'arrel, separades per punts. Fixeu-vos que l'ordre és invers al del sistema de fitxers UNIX/Linux. A més, fixeu-vos que un FQDN sempre acaba en un punt, ja que aquest és l'identificador del node arrel. Per les persones, un FQDN representa un nom (p.ex. d'una màquina). Pel sistema DNS, un FQDN representa un index per referenciar (i accedir a) un registre sobre un cert recurs; un **“resource record”** o **RR**. Per exemple, el node amb FQDN `www.eetac.upc.edu.` podria tenir associat un **RR** que contingúés “`147.83.96.2`”, que correspondria a l'adreça IP del servidor web de l'escola.



Als nodes immediatament per sota del node arrel, se'ls coneix com **Top-Level Domains** (TLDs). Als que estan a dos salts de l'arrel, **Second-Level Domains**, etc. Com es veurà, els TLDs solen indicar el tipus d'entitat o organització, o la seva ubicació geogràfica. I, els de segon nivell, el nom d'una organització (empresa, universitat, fundació, ONG, etc..) com *google*, *upc*, *intermon* o *wikipedia*.

2.2 RESOURCE RECORDS (RRs)

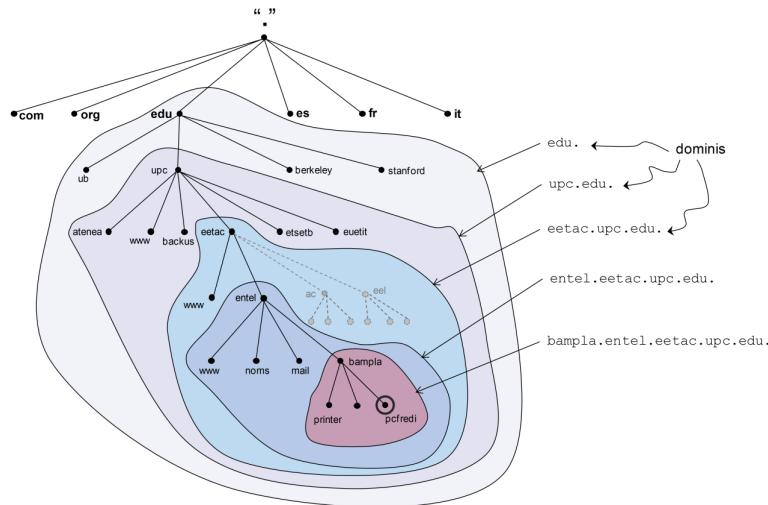
Els **RRs** no només poden contenir adreces IP. Habitualment, els nodes "fulla" (sense descendents), contenen RRs tipus A, que emmagatzemem una adreça IPv4. Els nodes interiors poden contenir qualsevol tipus de RR, adreces incloses. Hi ha RRs de varíes **classes**: Internet (**IN**), Chaosnet (**CH**) i Hesiod (**HS**), tot i que la classe Internet és pràcticament la única que es fa servir. A cada classe, existeixen diferents **tipus** de RRs segons la informació que contenen i la seva finalitat. Alguns RRs contenen informació relativa a hosts, d'altres relativa als serveis que una organització pot oferir i, d'altres, informació per a que el propi sistema DNS pugui funcionar. A cada **tipus**. En una analogia amb el sistema de fitxers de



arbre de directoris i, els RRs, a diferents fitxers emmagatzemats a cada directori (FQDN). Quan, per un cert node, hi ha més d'un RR del mateix tipus, se sol parlar d'un “**RR-set**”. La figura il·lustra la idea.

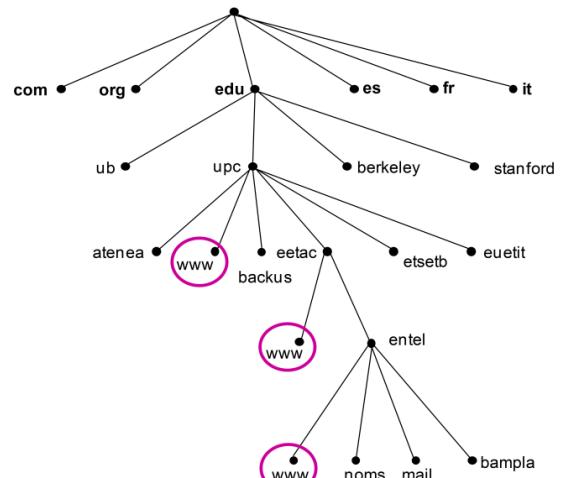
2.3 DOMINIS I SUB-DOMINIS

El terme “domini” apareix sovint en el camp de les xarxes i la telemàtica. En el context del DNS, s’entén per domini a un sub-arbre de l’espai de noms, és a dir, el conjunt de tots els nodes que són fills (descendents) d’un cert node “pare”. El nom d’un domini, és el nom de domini del node pare. Es diu que un domini és un sub-domini d’un altre, si el primer forma part del segon. Així, el domini `.edu.` és tot el sub-arbre per sota del node `.edu`. El domini `.upc.edu.`, és el sub-arbre per sota del node `upc.edu`. i és un sub-domini del domini `.edu.`, com es mostra a la següent figura. El domini `eetac.upc.edu.` engloba tots els nodes per sota del node `eetac.upc.edu`. (i, per tant, tots els noms que acaben en `eetac.upc.edu`). Aquest domini és un sub-domini de `upc.edu` i de `.edu`. Per exemple, el node `pcfredi.bampla.entel.eetac.upc.edu.` pertany als dominis: `bampla.entel.eetac.upc.edu`, `entel.eetac.upc.edu`, `eetac.upc.edu`, `upc.edu` i `.edu`.



2.4 AVANTATGES DE L'ESPAI DE NOMS DEL DNS

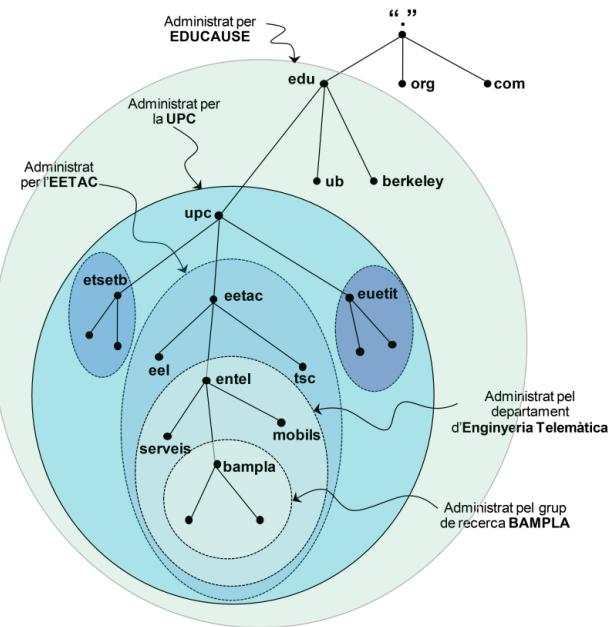
L’estructura de l’espai de noms anterior elimina els problemes del sistema basat en el fitxer `HOSTS.TXT`. Per una banda, el fet que l’espai de noms sigui jeràrquic permet partir-lo i gestionar els diferents (sub)dominis de forma independent. És a dir, la gestió dels dominis es pot **delegar** i no cal que sigui centralitzada. D’altra banda l’espai de noms jeràrquic del DNS també resol el problema de les col·lisions o duplicacions de noms. Si es garanteix que tots els nodes “germans” (fills del mateix pare) tinguin etiquetes diferents, aleshores no hi pot haver dos nodes amb el mateix nom (FQDN). És a dir, la unicitat local garanteix la unicitat global. D’aquesta manera, dos nodes poden tenir la mateixa etiqueta sense causar ambigüïtats. La figura mostra un exemple on el departament de telemàtica (entel), l’EETAC i la UPC poden tenir un node amb etiqueta `www` per referir-se als seus servidors web sense que hi hagi conflicte perquè els FQDNs són diferents: `www.entel.eetac.upc.edu`, `www.eetac.upc.edu`, `www.upc.edu`. I, per tant, no hi haurà ambigüïtats a l’hora de resoldre, per exemple, l’adreça IP de cada servidor, ja que cada node permetrà accedir a un RR tipus A diferent.



2.5 DELEGACIÓ DE DOMINIS

El concepte de **delegació** és fonamental al sistema DNS. Com s'ha comentat, una organització que administra un domini pot partit-lo en subdominis i **delegar-ne** la gestió (l'**autoritat**, en terminologia DNS) a altres organitzacions. És a dir, fer que se n'encarreguin d'afegir o treure nodes (noms) segons els convingui i que en mantinguin les dades (els RRs). A la vegada, una organització a qui es delega un domini pot definir tants subdominis com vulgui i delegar-ne l'autoritat; i així successivament.

Per exemple, el TLD `.edu` està gestionat per una organització anomenada EDUCAUSE, que delega subdominis a diferents universitats i centres educatius. Aquesta delega el subdomini `upc.edu` a la Universitat Politècnica de Catalunya. La UPC pot gestionar tota la branca `upc.edu.com` li convingui. Podria, per exemple, particionar el domini segons les diferents escoles (EETAC, EUETIT, ETSETB...) i delegar-ne la gestió. Així, l'EETAC podria afegir tots els noms que li convingués per sota de `eetac.upc.edu`. (p.ex. per anomenar màquines de laboratori) i associar-hi els RRs que calgués (p.ex. adreces IP). També podria gestionar tots els noms del domini `eel.eetac.upc.edu`, o `tsc.eetac.upc.edu`, corresponents a diferents departaments, però delegar el domini `entel.eetac.upc.edu` al departament de Telemàtica. Aquest darrer podria delegar el subdomini “`bampla`” al grup de recerca de Xarxes i Serveis de Banda Amplia.

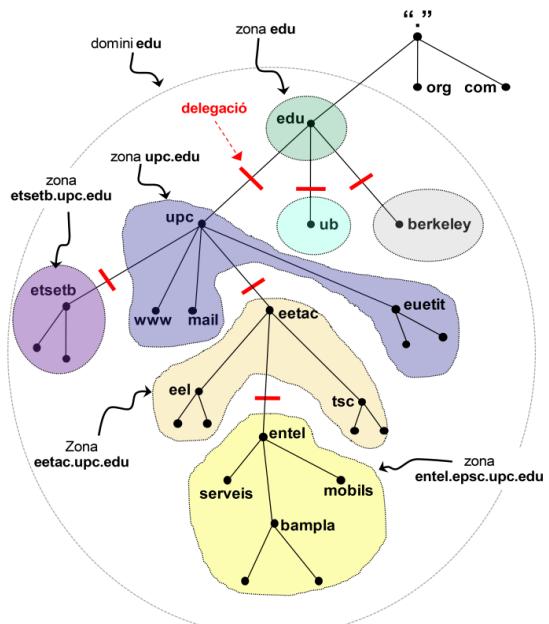


2.6 SERVIDORS DE NOMS I FITXERS DE ZONA

Els programes que emmagatzemem la informació DNS (**noms de dominis i RRs**) s'anomenen servidors de noms, servidors DNS, o **nameservers**. Aquests contenen la informació completa sobre “una part” de l'espai de noms, el que es coneix com una **zona**, que obtenen d'un **fitxer de zona** o d'un altre *nameserver*. Quan un servidor conté informació sobre una zona en un fitxer de zona, es diu que té l'autoritat sobre aquella zona. En aquests termes, delegar vol dir cedir l'autoritat sobre una zona a un altre servidor, de la mateixa organització o no.

A la figura, podeu veure l'espai de noms que cobreixen diferents zones i la idea de delegació com a partició de l'espai de noms. Fixeu-vos que, com les zones estan acotades per la delegació, aquestes no contenen informació que ha estat delegada. Per tant, el fitxer de zona de “`upc.edu`” conté tots els noms i RRs del domini `upc.edu`, excepte els dels dominis `etsetb.upc.edu` i `eetac.upc.edu` que s'han delegat. A l'exemple el domini EUETIT no han estat delegat.

És important entendre la diferència entre zona i domini. El terme domini es refereix a la organització lògica de l'espai de noms, mentre que una zona és un concepte administratiu. Una zona és un tros de l'espai de noms contigu sobre el qual un servidor té la informació completa i l'autoritat. Així, el node `eetac.upc.edu` pertany al domini `upc.edu` malgrat estar en una zona (fitxer) diferent. De la mateixa forma, el node `mobils.entel.eetac.upc.edu` pertany als dominis `eetac.upc.edu` (i `upc.edu`, i `edu`) tot



i estar a una zona diferent. Fixeu-vos que l'única informació que ha de tenir el servidor "autoritatius" de la zona **upc.edu** sobre el subdomini **eetac.upc.edu** és a quin servidor se n'ha delegat l'autoritat; un "**punter**" o **referència**. Aquesta referència ve donada pels RRs tipus **NS** (*Name Server*). Finalment, cal dir que no tots els servidors tenen fitxers de zona. És a dir no tots els servidors tenen autoritat sobre algun "tros" de l'espai de noms del DNS, com es veurà més endavant.

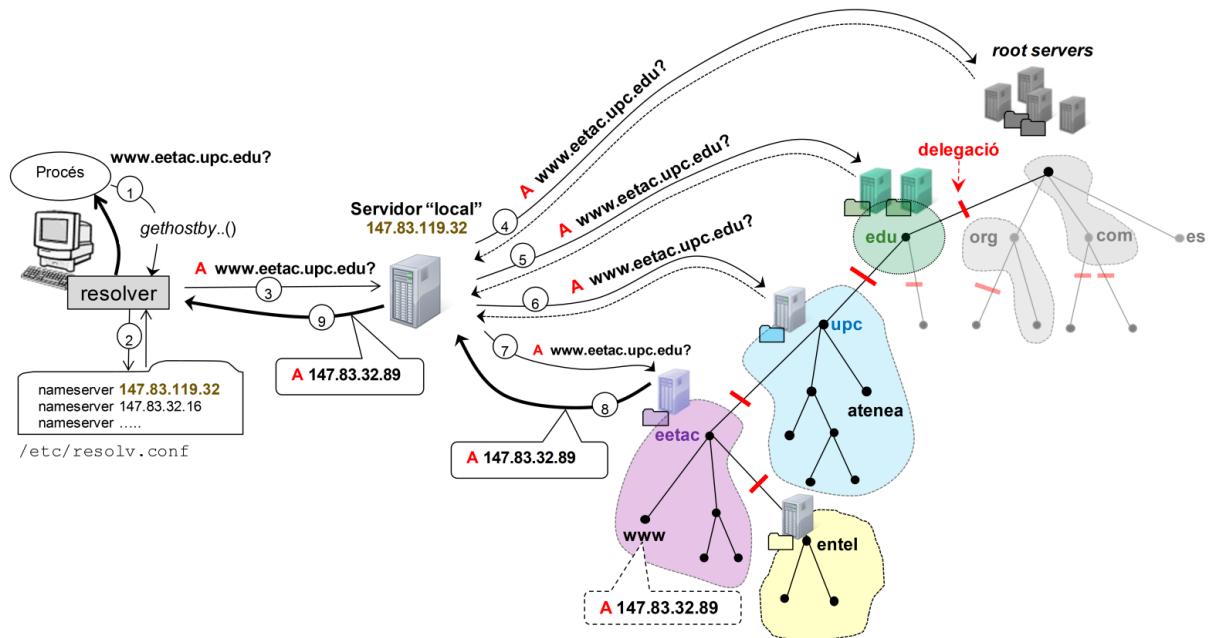
Les tres funcions principals d'un servidor de noms són:

- Mantenir fitxers de zona (que algun administrador ha confeccionat)
- Respondre a peticions sobre RRs d'altres *nameservers* o bé de *resolvers*
- Fer peticions a altres servidors.

En terminologia DNS, un *nameserver* fa el que es coneix com el procés de **resolució**. A partir d'un nom de domini (Per exemple **www.eetac.upc.edu**), d'una classe (per exemple "IN", d'Internet) i d'un tipus de RR (per exemple "A", d'adreça), retorna el RR associat, per exemple **147.83.32.89**. Si un servidor **no** té la informació que se li pregunta, o bé la demana a un altre servidor, o bé indica on es pot trobar (amb un RR tipus **NS**), o bé dona un error; com per exemple **NXDOMAIN** ("no such name") si el nom no existeix.

2.7 EL FUNCIONAMENT DEL DNS AMB UN EXEMPLE

Per veure la seqüència completa d'una petició DNS, suposeu que esteu asseguts davant de l'ordinador i, a la finestra d'un navegador, tecleieu **www.eetac.upc.edu** per accedir al servidor web de l'EETAC. Com sap el navegador (o qualsevol procés) quina és l'adreça IP corresponent? Doncs, amb el DNS. Ara bé, el navegador no interactua amb cap servidor DNS. Aquesta tasca la fa el *resolver* del vostre PC, que fa d'interfície entre les aplicacions (que no "parlen" DNS) i el sistema DNS. Típicament, les aplicacions fan servir alguna crida a sistema de l'API de *sockets* (que heu vist a l'assentatura de *Sistemes Operatius*), com per exemple **gethostbyname()**, per interrogar el resolver, passant-hi el nom que es vol resoldre, com s'indica a la figura. La seqüència seria la següent:

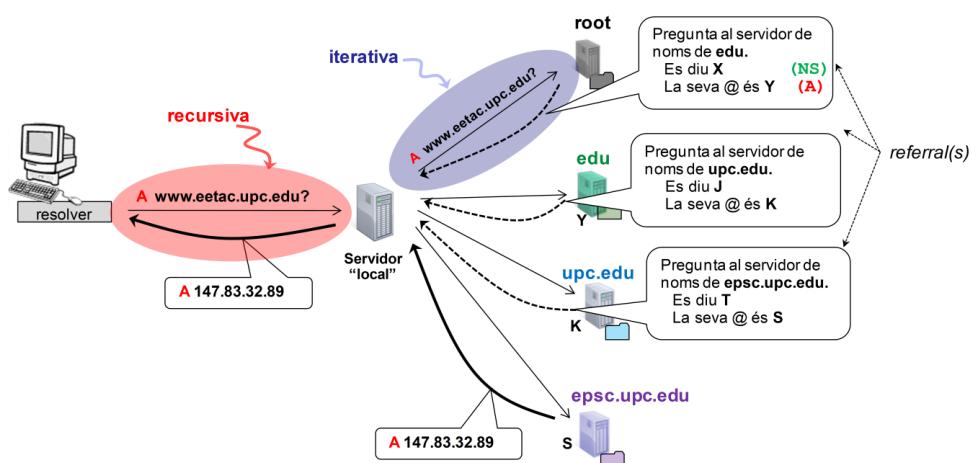


1. El procés (navegador) fa una petició per l'adreça de "www.eetac.upc.edu" al *resolver*.
2. Com sap el *resolver* a qui servidor preguntar? Tots els sistemes operatius ofereixen la possibilitat d'especificar una llista de servidors DNS. En el cas de Linux, existeix el fitxer **/etc/resolv.conf** que conté una llista de servidors (adreces IP) a qui el *resolver* pot consultar. El fitxer permet especificar més d'un servidor (la seva adreça) de manera que si un servidor no respon, el *resolver* envia la petició a un altre.
3. Sabent, doncs, a qui consultar, el *resolver* faria una petició DNS pel(s) RR(s) tipus **A** associats a **www.eetac.upc.edu**. a un servidor "local" especificat a **/etc/resolv.conf**. Suposem que aquest servidor no té l'autoritat sobre el domini **eetac.upc.edu**. Què faria? A qui consultaria? Per on començaria?

4. A l'arquitectura DNS hi ha un conjunt de servidors "arrel" (**root servers**) que contenen les adreces i noms dels servidors autoritatius dels TLDs. El servidor local reenviaria la pregunta a un dels servidors arrel. Aquest servidor, miraria el seu fitxer de zona i, en veure que `www.eetac.upc.edu.` no hi apareix, respondria dient que no té la resposta. Ara bé, com que el servidor coneix quin és el servidor del TLD `.edu.`, aquest l'indicaria a la resposta.
5. D'aquesta manera, el servidor local podria preguntar per `www.eetac.upc.edu.` al servidor autoritatius del domini `.edu.` Com aquest servidor hauria delegat tot el domini `upc.edu.` a la Universitat (i, per tant, no tindria la resposta), indicaria que no sap la resposta però que aquesta es pot trobar preguntant algun dels servidors de la UPC, la identitat (i adreces) dels quals inclouria a la resposta.
6. Així, el servidor local podria reenviar la pregunta al servidor de la UPC. Donat que tot el domini `eetac.upc.edu.` s'hauria delegat a l'EETAC, el servidor respondria amb els noms i adreces dels servidors de l'EETAC.
7. Finalment, el servidor local interrogaria a algun dels servidors de l'EETAC.
8. El servidor de l'EETAC preguntat, sí que tindria aquesta informació i la retornaria.
9. En rebre la resposta, el servidor local la reenviaria al *resolver*.
10. El *resolver* (activat mitjançant la crida d'una funció), retornaria la resposta, de manera que l'aplicació ja coneixeria l'adreça IP de `www.eetac.upc.edu.` i podria, per tant, enviar-hi tràfic. (En el cas d'un servidor http (web), establiria una connexió TCP pel port 80, com es veurà més endavant a l'assignatura.)

2.8 TIPUS DE CONSULTES DNS. RRs TIPUS NS

El sistema (i protocol) DNS permet que les entitats DNS (*nameservers* i *resolvers*) realitzin 2 tipus de consultes. Per una banda, hi ha les peticions **recursives**. En una petició recursiva, el servidor que la rep sempre retorna la resposta (si la sap) o un error. Si cal, per retornar la resposta, fa consultes a altres servidors, que poden ser recursives o no. D'altra banda, hi ha les peticions **iteratives** o **no recursive**s. Quan un servidor rep una petició iterativa, si sap la resposta (un RR), la dona. Si no, indica que no la sap, però que aquesta es pot trobar en un altre servidor. Fa el que es coneix com un **referral**, amb què referencia (amb un RR tipus NS) la identitat d'un altre servidor de noms "més proper" a la resposta. A l'exemple anterior, es donaven els dos tipus de consultes, com mostra la figura. Com que el servidor local rebia una petició recursiva, feia tantes peticions com calgués per donar la resposta.

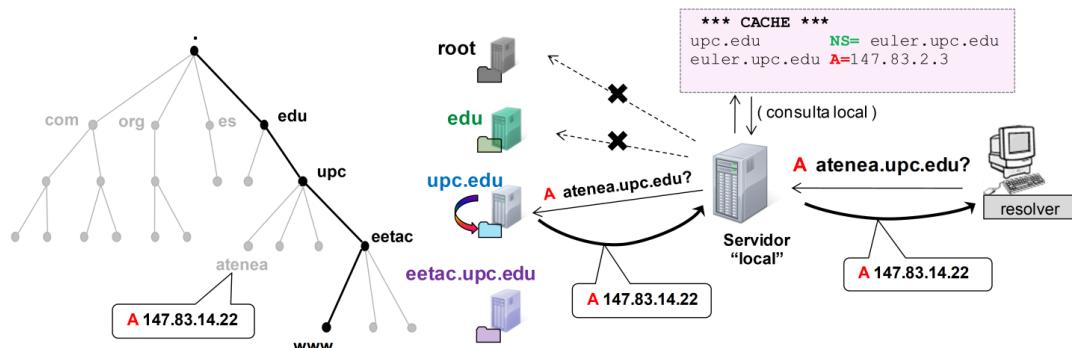


El tipus de petició s'indica en un flag en els missatges de consulta. En el 99.9% dels casos, els *resolvers* són "stub resolvers", funcions molt simples del sistema operatiu incapaces de seguir *referrals*. Per tant, fan peticions **recursives**. D'altra banda, els servidors de noms solen fer consultes **iteratives**. Com que les consultes recursives consumeixen més CPU i ample de banda, la majoria de servidors DNS es poden configurar, si cal, per denegar aquest tipus de consultes. Típicament, els servidors "locals" permeten consultes recursives, mentre que els servidors autoritatius per un determinat domini les deneguen. Alguns estudis mostren que els servidors arrel poden rebre de l'ordre de desenes de milers de peticions per segon. Per això, aquests servidors també solen denegar la recursió.

2.9 CACHING I TTLs

Segons l'explicat, cada consulta que féssim a un servidor sobre un nom de domini sobre el que no tingués l'autoritat seria reenviada a un servidor arrel, el qual, possiblement, referenciaria a un servidor d'un Top-Level Domain, i així successivament fins trobar un servidor que donés la resposta. Clarament, això retardaria molt el temps de resolució. Per tal de reduir aquest temps i la càrrega als servidors arrel (que rebrien cada petició que es fés al món!), els servidors DNS mantenen una *cache* de RRs (i, de fet, els *resolvers* també). La idea és que, quan un servidor processa una petició recursiva, aprèn molta informació. Cada vegada que és referit a una llista de servidors de noms "més propers" a una resposta, aprèn els servidors autoritatius per una determinada zona/domini (és a dir RRs tipus NS) i les seves adreces (RRs tipus A). A més, si el servidor autoritatius d'un domini li indica que un cert nom en la zona no existeix (NXDOMAIN), també ho recorda. Això es coneix com *negative-caching*. La idea darrera del *caching* és aprofitar tota aquesta informació (noms i RRs) per resoldre peticions futures.

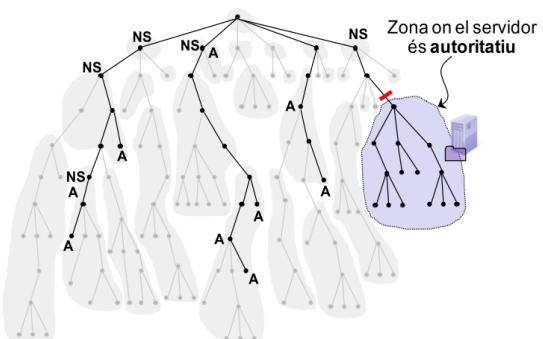
Per exemple, a l'escenari anterior, en preguntar per "www.eetac.upc.edu." el servidor de noms "local" hauria après entre d'altres, el nom i l'adreça del servidor de noms de la UPC (`euler.upc.edu.`, amb IP 147.83.2.3). Si, després, algú altre (no necessàriament el mateix resolver) preguntés per `atenea.upc.edu.`, el servidor local, consultant la cache, detectaria que pertany al domini `upc.edu.` de qui en coneix el servidor de noms. Per tant, no li caldria preguntar als servidors arrel pels servidors del TLD `.edu.` ni, després, preguntar a algun d'aquests pel servidor de `.upc.edu.`; com es mostra a la figura.



Com és lògic, el *caching* millora molt l'eficiència del DNS, ja que fa que la informació (FQDNs i els RRs associats) es vagi propagant arreu i estigui disponible a servidors "propers" als usuaris. A més, redueix enormement la càrrega als servidors de noms i als enllaços. Tot i això, és important veure que els servidors DNS **no** guarden indefinidament els RRs que aprenen. Si ho fessin, acabarien emmagatzemant tota la base de dades! A més, mai no aprendrien els canvis que es fessin en els fitxers de zona (per exemple, si la adreça IP d'un cert servidor canviés).

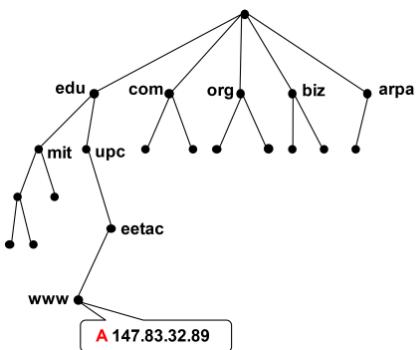
Al DNS, cada RR té associat un temps de vida (*time-to-live*, **TTL**), passat el qual un servidor **no** està autoritzat a guardar-lo més i l'ha de descartar. En aquest sentit, hi ha un compromís **consistència-rendiment** en la tria dels TTLs. TTLs "grans" tendeixen a reduir el temps de resolució (ja que els RR "viuen" més temps allà on són apresos), però poden fer que els servidors responguin amb dades obsoletes. Per contra, amb TTLs petits, els canvis en els RRs es propaguen ràpid, però el temps de resolució i la càrrega tendeix a augmentar.

Així, amb el *caching* i el TTL, la visió que un servidor DNS podria tenir sobre el contingut de la base de dades ens la podríem imaginar com la de la figura. A banda de tota la informació sobre les zones on fos autoritatius, aquest podria tenir a la cache "trossos" de l'arbre de noms i els RRs associats (marcats en negre a la figura). Noves peticions de resolució farien que apareguéssin noves branques, mentre que altres branques podrien desaparèixer quan el seu TTL expirés. Una conseqüència del caching és, per tant, que un servidor pot servir (responder) RRs de zones sobre les que **no** té autoritat; és a dir sobre noms que no apareixen en el seu fitxer de zona. Es diu que un servidor pot donar respostes "**no-autoritatives**". Finalment, com s'ha comentat, hi ha servidors que no tenen fitxers de zona. Aquests servidors fan només de caché i se'ls anomena **caching-only nameservers**.



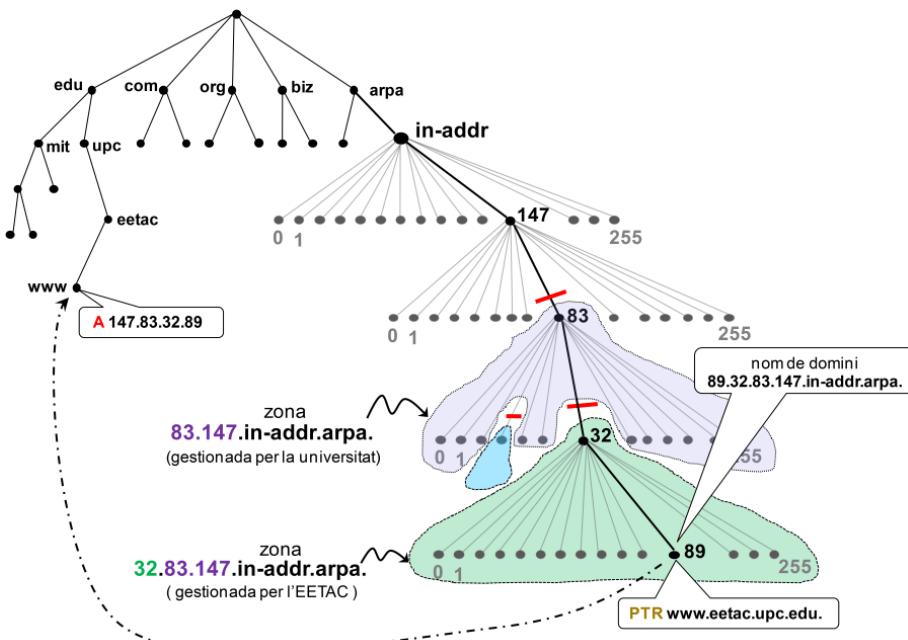
2.10 RESOLUCIÓ DE NOMS. RRs TIPUS PTR

Fins ara hem vist com el DNS pot resoldre adreces. És a dir, a partir d'un nom, trobar-ne l'adreça IP. De vegades, però, interessa conèixer el nom associat a una certa adreça, per exemple 147.83.32.89. Una possible manera de donar resposta a aquest tipus de peticions podria ser fer una cerca a l'espai de noms fins a trobar el RR tipus A amb contingut 147.83.32.89; el FQDN del node on ens aturéssim, seria la resposta. Aquesta solució requeriria una cerca exhaustiva per l'espai de noms que podria ser molt lenta a mesura que el nombre de nodes cresqués. A més, l'espai de noms pot estar repartit per molts servidors de manera que aquesta solució seria molt ineficient. El problema és que les adreces, com 147.83.32.89, no tenen res a veure amb les etiquetes dels nodes i, per tant, no es poden cercar de forma eficient. Quina solució adopta el DNS? Doncs, ben senzilla: habilitar un domini on les etiquetes dels nodes **sí** que tinguin alguna relació amb les adreces. I no només això sinó que s'aprofita l'estructura jeràrquica de les adreces IP. Aquest domini és el domini `in-addr.arpa`.



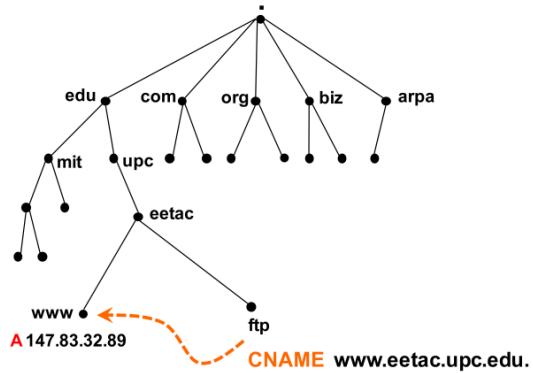
La idea consisteix en fer servir cada octet d'una adreça IPv4 com a etiqueta d'un node, començant pel més significatiu, i d'un RR d'un tipus especial (`PTR`). Així, per cada IP que vulguem, existeix un node a l'espai de noms. Per exemple, per l'adreça 147.83.32.89, el FQDN del seu node seria `89.32.83.147.in-addr.arpa`. I, el RR tipus PTR associat podria contenir el nom del host, `www.eetac.upc.edu.`, com es mostra a la figura. D'alguna manera aquest RR "apunta" al node amb FQDN `www.eetac.upc.edu`; d'aquí que, a aquests RRs, se'ls anomeni `PTR`, de PoinTeR.

Aquest esquema permet trobar el nom associat a una adreça (com 147.83.32.89), simplement invertint l'ordre dels octets de l'adreça, i cercant dins del domini `in-addr.arpa`. A més, com els octets més significatius corresponen a nodes a posicions més altes de l'arbre de noms, s'aprofita la jerarquia de l'adreçament IP, de manera que es poden delegar subdominis de la branca `in-addr.arpa` a diferents organitzacions. Per exemple, com la UPC posseeix tot el rang de classe B 147.83.0.0/16, se li pot delegar tot el domini `83.147.in-addr.arpa`. I aquesta podria, per exemple, delegar el subdomini `32.83.147.in-addr.arpa` a l'escola (suposant que l'EETAC disposés de tot el rang 147.83.32.0/24). Fixeu-vos que, en aquest esquema, només es poden delegar dominis corresponents a rangs /8, /16 i /24 ja que va ser ideat quan l'encaminament a Internet estava basat en classes. A la secció 1.26, s'explica un mecanisme emprat per poder delegar rangs arbitraris, com ara /27 o /29. Aquest mecanisme empra RRs del tipus CNAME, que es veuen a continuació.



2.11 ÀLIES I NOMS CANÒNICS. RRS TIPUS CNAME

Sovint interessa que una mateixa màquina sigui accessible amb diferents noms. Moltes vegades, una mateixa màquina allotja diferents serveis, com per exemple un servidor web, un ftp o un servidor de correu, però interessa accedir a la màquina amb un nom que ens recordi el servei o l'aplicació en particular (com www, ftp, mail o smtp). Per adreçar aquesta necessitat, el DNS defineix un tipus especial de RR, el **CNAME**, que permet crear àlies; és a dir, noms alternatius a un cert nom de domini existent, el **nom canònic**. A l'exemple, s'ha creat l'àlies **ftp** pel nom canònic **www.eetac.upc.edu**. La petició d'un RR per **ftp.eetac.upc.edu** fa que el servidor cerqui aquell RR pel nom **www.eetac.upc.edu**, excepte si es demana precisament per un RR de tipus **CNAME**.



2.12 TIPUS DE RESOURCE RECORDS

Existeixen varis tipus de RR a banda dels **A**, **NS**, **PTR** i **CNAME** que s'han comentat. Alguns apareixen a l'especificació origina del DNS. D'altres, s'han afegit amb noves propostes i usos del sistema. I, d'altres han quedat obsolets i no es fan servir. La següent taula resumeix el nom dels RRs enregistrats al IANA (*Internet Assigned Numbers Authority*), el seu codi numèric i el RFC on s'han proposat. Alguns d'aquests RRs els veurem en aquest capítol, d'altres a capítols posteriors i d'altres no els veurem a l'assignatura, però és possible que els veieu a altres assignatures (p.ex. els que tenen a veure amb el *DNSSEC*). Els marcats negreta són dels que més s'utilitzen i que veurem a l'assignatura.

TYPE	Value and meaning	Reference
✓ A	1 a host address	[RFC 1035]
✓ NS	2 an authoritative name server	[RFC 1035]
MD	3 a mail destination (Obsolete - use MX)	[RFC 1035]
MF	4 a mail forwarder (Obsolete - use MX)	[RFC 1035]
✓ CNAME	5 the canonical name for an alias	[RFC 1035]
✓ SOA	6 marks the start of a zone of authority	[RFC 1035]
MB	7 a mailbox domain name (EXPERIMENTAL)	[RFC 1035]
MG	8 a mail group member (EXPERIMENTAL)	[RFC 1035]
MR	9 a mail rename domain name (EXPERIMENTAL)	[RFC 1035]
NULL	10 a null RR (EXPERIMENTAL)	[RFC 1035]
WKS	11 a well known service description	[RFC 1035]
✓ PTR	12 a domain name pointer	[RFC 1035]
HINFO	13 host information	[RFC 1035]
MINFO	14 mailbox or mail list information	[RFC 1035]
✓ MX	15 mail exchange	[RFC 1035]
✓ TXT	16 text strings	[RFC 1035]
RP	17 for Responsible Person	[RFC 1183]
AFSDB	18 for AFS Data Base location	[RFC 1183] [RFC 5864]
X25	19 for X.25 PSDN address	[RFC 1183]
ISDN	20 for ISDN address	[RFC 1183]
RT	21 for Route Through	[RFC 1183]
NSAP	22 for NSAP address, NSAP style A record	[RFC 1706]
NSAP-PTR	23 for domain name pointer, NSAP style	[RFC 1348]
SIG	24 for security signature	[RFC 4034] [RFC 3755] [RFC 2535]
KEY	25 for security key	[RFC 4034] [RFC 3755] [RFC 2535]
PX	26 X.400 mail mapping information	[RFC 2163]
GPOS	27 Geographical Position	[RFC 1712]
✓ AAAA	28 IPv6 Address	[RFC 3596]
LOC	29 Location Information	[RFC 1876]
NXT	30 Next Domain - OBSOLETE	[RFC 3755] [RFC 2535]
EID	31 Endpoint Identifier	
NIMLOC	32 Nimrod Locator	
✓ SRV	33 Server Selection	[RFC 2782]
ATMA	34 ATM Address	
✓ NAPTR	35 Naming Authority Pointer	[RFC 2915] [RFC 2168] [RFC 3403]
KX	36 Key Exchanger	[RFC 2230]
CERT	37 CERT	[RFC 4398]
A6	38 A6 (Experimental)	[RFC 3226] [RFC 2874]
DNAME	39 DNAME	[RFC 2672]
SINK	40 SINK	[Eastlake]
OPT	41 OPT	[RFC 2671]
APL	42 APL	[RFC 3123]
DS	43 Delegation Signer	[RFC 4034] [RFC 3658]
SSHFP	44 SSH Key Fingerprint	[RFC 4255]
IPSECKEY	45 IPSECKEY	[RFC 4025]
RRSIG	46 RRSIG	[RFC 4034] [RFC 3755]
NSEC	47 NSEC	[RFC 4034] [RFC 3755]
DNSKEY	48 DNSKEY	[RFC 4034] [RFC 3755]
DHCID	49 DHCID	[RFC 4701]
NSEC3	50 NSEC3	[RFC 5155]
NSEC3PARAM	51 NSEC3PARAM	[RFC 5155]

Unassigned	52-54	
HIP	55 Host Identity Protocol	[RFC 5205]
NINFO	56 NINFO	
RKEY	57 RKEY	
TALINK	58 Trust Anchor LINK	
Unassigned	59-98	
✓ SPF	99 Sender Policy Framework	[RFC 4408]
UINFO	100	[IANA-Reserved]
UID	101	[IANA-Reserved]
GID	102	[IANA-Reserved]
UNSPEC	103	[IANA-Reserved]
Unassigned	104-248	
TKEY	249 Transaction Key	[RFC 2930]
TSIG	250 Transaction Signature	[RFC 2845]
✓ IAXFR	251 incremental transfer	[RFC 1995]
✓ AIXFR	252 transfer of an entire zone	[RFC 1035] [RFC 5936]
MAILB	253 mailbox-related RRs (MB, MG or MR)	[RFC 1035]
MAILA	254 mail agent RRs (Obsolete - see MX)	[RFC 1035]
*	255 A request for all records	[RFC 1035]
Unassigned	256-32767	
TA	32768 DNSSEC Trust Authorities	
DLV	32769 DNSSEC Lookaside Validation	[RFC 4431]
Unassigned	32770-65279	
Private use	65280-65534	
Reserved	65535	

2.13 ENCaminament de correu electrònic. RRs tipus MX

Una de les aplicacions més utilitzades del sistema DNS és l’“encaminament” del correu electrònic. De fet, sense el DNS, el servei de correu electrònic com funciona avui dia no seria possible.

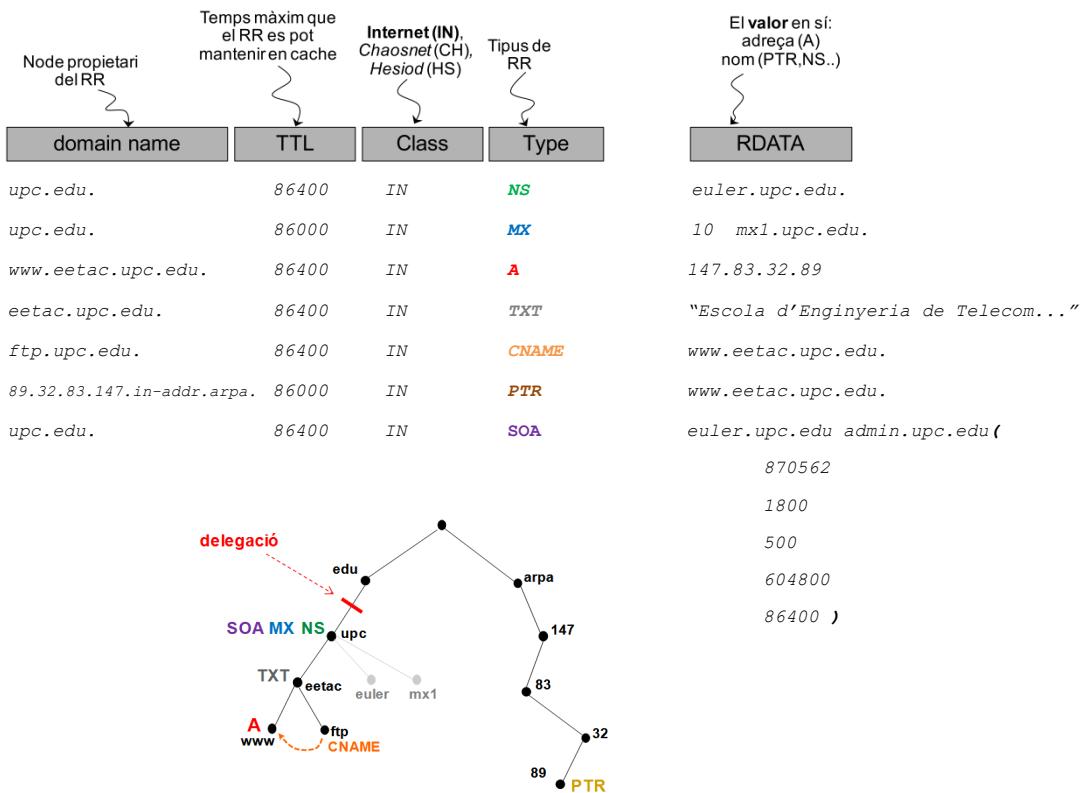
Suposeu que teniu un compte de correu a `hotmail.com` (per exemple `pepito@hotmail.com`). Quan algú us envia un correu, com és que el rebeu encara que tingueu el vostre PC apagat? La resposta és ben senzilla: els correus que els vostres amics us envien no l’envien al vostre PC sinó que l’envien a un servidor de correu on hi teniu configurada una **bústia** (en aquest exemple, anomenada `pepito`). D’aquesta forma, quan voleu accedir al vostre correu, el vostre client de correu accedeix a la vostra bústia en el servidor i es descarrega al vostre PC els e-mails allà guardats. Per a aquest propòsit es fan servir protocols com el POP3 o l’IMAP4. El vostre client sap a quin servidor i bústia ha d’accedir perquè aquesta informació s’ha especificat via configuració. De fet, el vostre client també té configurat un altre servidor de correu “sortint” a qui transfereix, mitjançant el protocol SMTP els correus que vosaltres envieu. La pregunta és, com arriben els correus a les vostres bústies? Dit d’una altra forma: com saben els servidors de correu “sortint” a quin servidor han d’enviar els correus que envieu? Aquí és on entra en joc el DNS.

El DNS permet associar a cada domini de correu (per exemple `hotmail.com`) un o més servidors de correu configurats per acceptar el correu destinat a les bústies del domini. Aquesta associació es fa configurant uns RRs especials anomenats *mail exchanger records* (RRs tipus `MX`) al domini en qüestió. Els RR tipus `MX` contenen el nom DNS del servidor de correu i un camp numèric que indica la prioritat o preferència en cas que el domini tingui més d’un RR tipus `MX`. Com més baix és aquest valor, més preferent és el servidor. Així, quan un servidor sortint ha d’enviar un correu, fa una consulta pels RRs tipus `MX` associats al domini del destinatari (el nom després de `@` a l’adreça de correu) per tal de descobrir els servidors del domini receptor i les seves adreces. En aquest sentit, fixeu-vos que el servidor DNS ha de tenir RRs tipus `A` associats als noms de servidors de correu.

2.14 FORMAT TEXTUAL DELS RRS

Els RRs segueixen un cert format textual en la seva especificació als fitxers de zona. Cada RR té 5 camps. El primer camp indica el nom de domini (relatiu o FQDN). El segon camp indica el TTL en segons, és a dir el temps màxim que un *nameserver* pot guardar el RR a la cache. El tercer camp és la classe i, el quart, el tipus de RR. Finalment, el cinquè camp, `RDATA`, conté el valor en sí del RR (una adreça, un nom de domini, un text, etc., dependent del tipus de RR). La figura mostra el format (textual) genèric d'un RR, alguns exemples de RRs i la representació gràfica a l'arbre de l'espai de noms que hem emprat fins ara.

Fixeu-vos que cada RR ocupa una línia, excepte el darrer, de tipus `SOA`, que se sol escriure en vàries línies fent servir parèntesis. Com es veurà, són precisament els RRs tipus `SOA` els que marquen qui té l'autoritat sobre un cert domini i, per tant, indiquen l'inici d'un domini delegat.



2.15 FORMAT DELS NOMS DE DOMINI

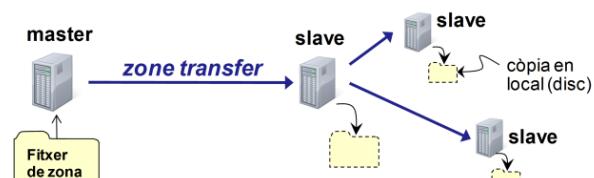
El sistema DNS imposa certes restriccions pel que fa a les etiquetes dels nodes i els noms de domini de l'espai de noms. Per exemple, els caràcters permesos per a una etiqueta estan restringits a **lletres** (A-Z), **dígits** (0-9) i **guions** (""). Amb la introducció dels anomenats *Internationalized Domain Names* (IDNs) es preveu que el conjunt de caràcters s'estengui força més. També hi ha les següents restriccions.

Primer, l'etiqueta associada a un node no pot tenir més de 63 caràcters. Segon, la longitud màxima d'un FQDN és de 127 etiquetes. És a dir, la profunditat màxima de l'arbre de noms és 127. Finalment, els RFCs 1034 i 1035 recomanen, per simplificar les implementacions, que la longitud d'un FQDN no excedeixi 255 caràcters.

2.16 TIPUS DE SERVIDORS DE NOMS

Si cau un servidor DNS, cap dels RRs (màquines, serveis) del seu domini serà accessible (potser no immediatament, degut al *caching*, però sí a la llarga). Els servidors DNS solen estar redundats per varis motius: per garantir el servei de resolució, per repartir la càrrega entre varis servidors (*load balancing*), per protegir servidors darrera d'un firewall i exposar ràpliques d'aquest, per minimitzar el temps de resolució, i perquè tant l'estàndar (RFC1034) com les organitzacions que gestionen l'arquitectura obliguen a tenir com a mínim 2 servidors per cada domini.

Per facilitar-ne la gestió, hi ha dos tipus de servidors DNS. Els **master** (o *primary master*, per una zona) llegeixen la informació de zona d'un fitxer de zona (confeccionat per un administrador). En canvi, els **slave** (o *secondary master*) obtenen la informació de zona d'un altre servidor autoritatiu, el seu "master server", que pot ser un altre servidor *slave*, com es mostra a la figura. El procediment pel qual un *slave* obté la informació de zona d'un altre servidor es coneix com **zone transfer** (AXFR). Habitualment, els servidors *slave* fan una còpia en disc de la informació de zona. Així, si cauen, no cal que iniciïn un altre *zone transfer*. Val a dir que tan autoritatiu (sobre un cert domini) és un servidor *master* com qualsevol dels seus esclaus. I que un servidor pot ser *master* per una zona i *slave* per una altra.



2.17 RRs TIPUS SOA

Fins ara hem vist que la informació que manega el DNS s'origina als anomenats fitxers de zona i que el sistema permet delegar porcions de l'espai de noms (i els corresponents RRs) a altres organitzacions (i, per tant, a altres servidors). És a dir, cedir-ne l'autoritat. Existeix un RR especial per indicar l'autoritat sobre un domini; l'anomenat *Start-of-Authority* o **SOA**. Tot fitxer de zona inclou un (i només un) **SOA**. El format d'un RR tipus **SOA** és com el que s'indica al següent exemple.

```

upc.edu. 86400 IN SOA euler.upc.edu. admin.upc.edu. (
    2010062801 ; serial (YYYYMMDDNN)
    6h           ; refresh every 6 hours
    1h           ; retry after 1 hour
    1w           ; expire after 1 week
    1h )         ; negative caching TTL of 1 hour
  
```

El camp **RDATA** d'un **SOA** conté el nom canònic del servidor master de la zona i, a més, l'adreça de correu electrònic de la persona responsable del domini/zona. A continuació, apareixen 5 camps (que se solen escriure com a la figura, i on els textos després dels punt-i-coma són comentaris).

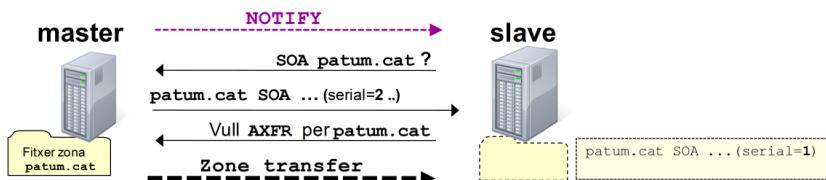
El primer camp, **serial**, és el número de sèrie del fitxer de zona. Indica com és d'actual el fitxer de zona. Quan un *slave* contacta un *master*, primer pregunta per aquest valor. Si el **serial** del *slave* és inferior, vol dir que les seves dades estan obsoletes i inicia una **transferència de zona**. Cada vegada que un administrador modifica el fitxer de zona, cal que fixi el **serial** a un valor més elevat, per a que els diferents servidors *slave* es baixin la nova versió del fitxer. Tot i que l'estàndar no fixa cap format per aquest camp, se sol fer servir el format YYYYMMDDNN, on Y correspon als dígits de l'any actual, M als del mes, D als del dia i N als d'un nombre natural.

La resta de camps són valors temporals. El camp **refresh** indica quant sovint un *slave* hauria d'interrogar el *master* de la zona (demanant el **SOA**) per comparar el **serial** d'un fitxer de zona i veure si aquest ha canviat. El camp **retry** té el següent significat. Si un *slave* no pot contactar amb el *master* després de **refresh** segons, intenta fer-ho cada **retry** segons. Típicament es tria **retry** < **refresh**. Si un *slave* no pot contactar amb el *master* després de **expire** segons, aquest expira la zona: no respon a RRs dins de la zona. Finalment l'últim camp ha canviat amb el temps. Antigament representava el TTL per defecte. A partir de l' RFC2308 es fa servir la directiva **\$TTL** pel *time-to-live* per defecte (que es veurà més endavant) i el valor en el **SOA** és el TTL per a respostes negatives (**NXDOMAIN**).

2.18 ALGUNES FUNCIONALITATS MÉS RECENTS DEL DNS

Des de l'especificació del DNS, s'han anat proposant millors i nous mecanismes al sistema per adaptar-lo a noves necessitats en el món IP i d'Internet. En aquesta secció es comenten breument algunes d'aquestes millors "post RFC1035".

La primera té a veure amb la sincronització de *slaves* i *masters*. El mecanisme original (basat en els temporitzadors **refresh** i **retry** que acabem de veure) podria ser massa lent en un entorn on els fitxers de zona canviessin força sovint. Per això, l' RFC 1995 defineix la operació **NOTIFY**, amb la qual un servidor *master* pot notificar de canvis a la seva zona a un *slave*, indicant la necessitat de consultar el **SOA** per comprovar el **serial** i sol·licitar una transferència de zona (operació **AXFR**) si cal. La figura següent il·lustra el procés per un fitxer de zona que inclou el domini **patum.cat**.



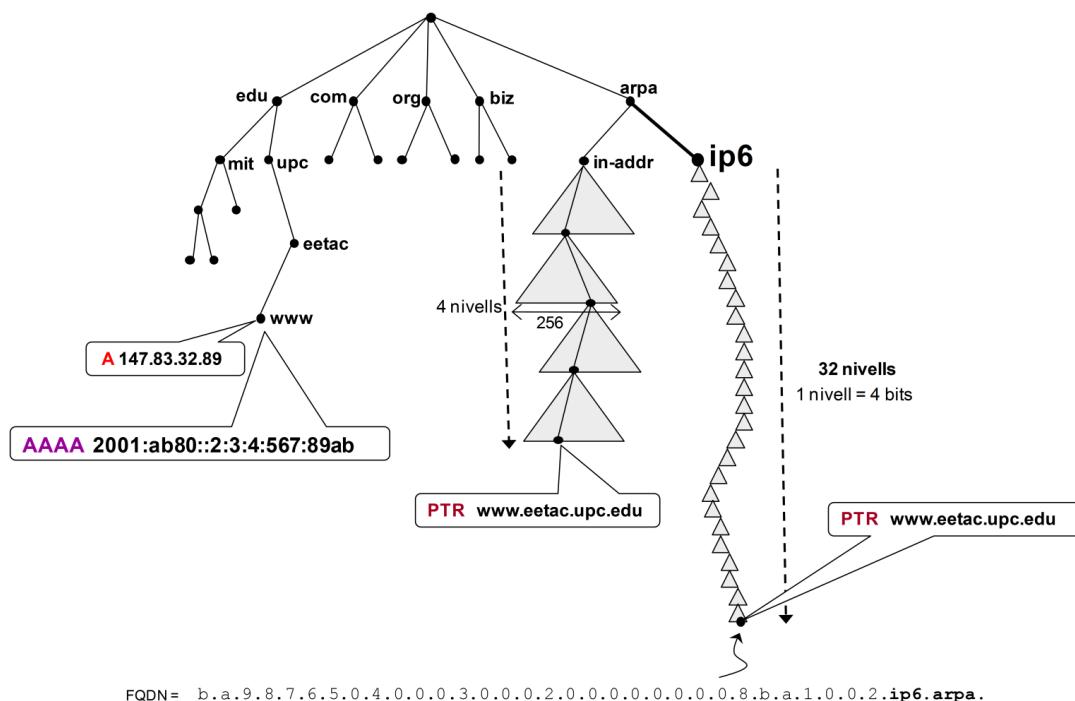
Per tal d'accelerar el procés, el mateix RFC defineix les anomenades transferències de zona incrementals (*incremental zone transfers*, **IXFR**), amb les quals només s'envia la informació al fitxer de zona que ha canviat des de la darrera transferència de zona (o valor del **serial**).

Finalment, una altra modificació important és l'anomenat DNS dinàmic (**dynamic DNS o DDNS**) especificat a l'RFC 2136. El DDNS permet fer canvis als fitxers de zona sense la supervisió de l'administrador. Defineix una nova operació, **UPDATE**, amb la qual un dispositiu (p.ex. un servidor DHCP) pot modificar RRs en el fitxer de zona. Per exemple, el RR tipus **A** per una màquina (nom) l'adreça IP de la qual depèn de l'assignació que li hagi fet un servidor DHCP. Fixeu-vos que les funcionalitats anteriors de **NOTIFY** i **IXFR** tenen especial sentit en un entorn com aquest.

2.19 DNS A XARXES IPv6. RRs TIPUS AAAA

Com s'ha vist, per IPv4 existeix un tipus de *resource-record* (RR) (el tipus **A**) que permet emmagatzemar una adreça IPv4 per un determinat nom de domini. Per tal d'ofrir la mateixa funcionalitat a IPv6, l' RFC 3596 defineix un nou tipus de RR per guardar adreces IPv6 de 128 bits. El nom d'aquest tipus de RR és **AAAA** i es llegeix "quad-A", ja que la seva longitud és quatre vegades la d'un tipus **A**. Així, si el servidor web de l'EETAC (amb nom de domini `www.eetac.upc.edu.`) tingués l'adreça IPv6 `2001:ab80::2:3:4:567:89ab`, l'especificació del corresponent *quad-A* a un fitxer de zona podria ser:

```
www.eetac.upc.edu. 86400 IN AAAA 2001:ab80:0000:0002:0003:0004:0567:89ab
```



A banda dels *quad-A*, es va proposar un altre RR (**I'A6**) que generalitzava els RRs per emmagatzemar no només adreces sinó prefixes. Actualment aquests RRs es consideren obsolets.

Com es resolen les adreces? Si ho recordeu, a IPv4, el problema es resol amb un domini especial (`in-addr.arpa.`), invertint l'ordre dels octets de l'adreça i fent servir un RR tipus `PTR`. La solució adoptada per IPv6 és conceptualment idèntica (fent servir RRs tipus `PTR`), excepte que es fa servir un domini diferent, `ip6.arpa`.

L'altra diferència és que, així com a IPv4 es defineix un node a l'espai de noms per cada octet (el que genera 4 nivells), a IPv6 es defineix un node per cada 4 bits, el que dóna fins a 32 nivells, amb 16 possibles nodes "fill" a cada nivell, com es mostra a la figura. Fixeu-vos que el FQDN al domini `ip6.arpa.` ha de contenir tots els bits. És a dir, no es poden ometre zeros per compactar el FQDN corresponent a l'adreça. Per tant, l'especificació a un fitxer de zona d'un RR tipus `PTR` per IPv6 podria ser:

```
b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.0.0.0.0.0.0.8.b.a.1.0.0.2.ip6.arpa IN PTR www.eetac.upc.edu
```

2.20 EL PROTOCOL DNS

Per realitzar consultes, el sistema DNS incorpora un protocol senzill del tipus petició-resposta. Habitualment, aquest protocol servir UDP com a protocol de transport (pel port 53), tot i que, en certs casos, fa servir el TCP (per exemple en les transferències de zona). L'especificació inicial del DNS limitava la mida dels missatges a 512 bytes. L'RFC 2671 (*Extension Mechanisms for DNS, EDNS0*) estén aquesta limitació a una mida negociable. Per ser eficient, la informació DNS va codificada amb un mecanisme senzill de compressió (que no veurem).

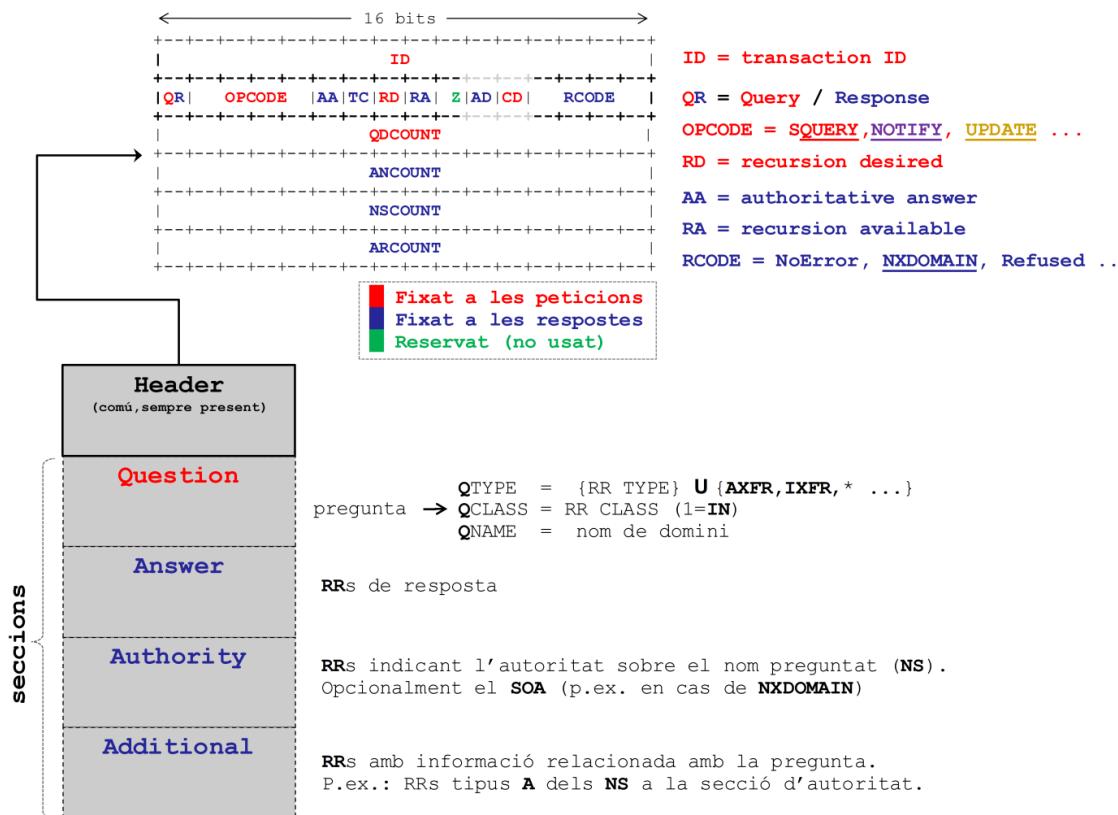
Cada petició DNS inclou, entre d'altres:

- una pregunta (és a dir quin RR es desitja per quin FQDN)
- si es vol recursió o no.

D'altra banda, les respostes inclouen:

- si se suporta recursió (flag RA, *recursion available*)
- si la resposta és autoritativa (flag AA, *authoritative answer*)
- un codi d'error / èxit
- la pregunta
- la resposta a la pregunta
- informació sobre l'autoritat del nom de domini preguntat
- i un camp d'informació addicional.

Les peticions i les respostes tenen el mateix format, que es resumeix a la figura següent. Cada missatge té una capçalera comuna (*header*) i 4 seccions opcionals. Els camps en vermell es fixen a les peticions, mentre que els camps en blau es fixen a les respostes.



La capçalera conté un identificador de transacció (**ID**) que permet relacionar les respostes amb les peticions. A continuació hi ha 16 bits de *flags* amb diferents significats. El bit **Q/R** indica si el missatge és una petició (**Query**) o una resposta (**Response**). El camp **OPCODE** (*operation code*) indica el tipus de petició com ara **SQUERY** (*standard query*), **NOTIFY** o **UPDATE** (el significat dels quals ja hem vist). El flag **RD** (*recursion desired*) indica si es vol recursió o no.

A les respostes, el flag **RA** (*recursion available*) indica si el servidor suporta recursió, mentre que el flag **AA** (*authoritative answer*) indica si la resposta és autoritativa o no. Finalment, el camp **RCODE** (*return code*) indica el resultat de la

operació. Aquest pot ser, per exemple, **NoError** (si no hi ha hagut errors), **NXDOMAIN** (si el nom no existeix) o **Refused**, si es denega la petició. Finalment, els camps **QDCOUNT**, **ANCOUNT**, **NSCOUNT** i **ARCOUNT** indiquen el nombre d'ítems a cadascuna de les seccions que van a continuació.

La primera secció inclou la pregunta i s'omple a les peticions. La pregunta té tres camps, on la **Q** ve “question”. El primer, **QTYPE**, indica o bé el tipus de RR que es demana, o bé un codi especial com **AXFR** (solicitud de *zone transfer*) o **IXFR** (per *incremental zone transfers*). El camp **QCLASS**, indica la classe (a efectes pràctics **IN**ternet). I, finalment, el **QNAME** indica un nom de domini.

La resta de seccions es fixen a les respostes. La secció **Answer** conté els RR de resposta. La secció **Authority** pot contenir RRs (p.ex. de tipus **NS**) indicant l'autoritat sobre el nom preguntat (p.ex. en cas de respostes no autoritatives), o bé un **SOA**, per exemple si la resposta és **NXDOMAIN**. Finalment, la secció **Additional** conté RRs amb informació addicional que pot facilitar trobar la resposta, com, per exemple, RRs tipus **A** amb les adreces dels servidors de noms (**NS**, indicats a la secció **Authority**) autoritatius del FQDN preguntat.

2.21 FORMAT DELS FITXERS DE ZONA

Els fitxers de zona són fitxers de text que descriuen una zona i que els servidors (*master*) llegeixen per carregar la informació sobre la zona. Tenen un format estàndard (especificat a l' RFC 1035) i indiquen:

- qui té l'autoritat sobre la zona i en quin punt de l'espai de noms comença (és a dir, un RR tipus **SOA**)
- noms de domini (hosts, serveis), i els RRs associats (adreces, ,etc.), en el format textual explícit
- informació de delegació de subdominis. És a dir, pels subdominis del domini principal de la zona que s'han delegat, a quins servidors de noms se n'ha cedit l'autoritat. En altres paraules, RRs tipus **NS**.

Els fitxers de zona sovint fan servir:

- noms *relatius* (a un cert node), que es distingeixen dels FQDN perquè no acaben amb un punt.
- abreviacions
- comentaris, que comencen amb un “;”
- **directives** (estàndar i no estàndar)

Les directives que més s'utilitzen són:

- **\$ORIGIN** (definida a RFC1305): conté la cadena que s'afegeix per a completar noms *relatius*.
- **\$INCLUDE** (RFC 1305) : permet incloure un fitxer. És l'equivalent a `#include <fitxer.h>` de C/C++
- **\$TTL** (RFC 2308): indica el TTL per defecte, pels RRs que no l'indiquen explícitament.
- **@**: s'expandeix amb el valor d' **\$ORIGIN**.

Per exemple, un fitxer de zona fent servir abreviacions, comentaris i directives podria ser el següent, pel domini **farmaciola.cat**. Fixeu-vos com alguns camps s'hereten dels RRs anteriors (p.ex. la classe **IN**), que **@** s'expandeix amb el valor de **\$ORIGIN**, és a dir, **farmaciola.cat**. i que el nom de **www.farmaciola.cat**. s'ha escrit com **www**; és a dir, relatiu a **farmaciola.cat**.

```
-----  
$ORIGIN farmaciola.cat.  
$TTL 6h ; sis hores  
@ IN SOA aspirina.farmaciola.cat. hostmaster.farmaciola.cat.  
      MX 10 mail ; classe és INternet  
  
ftp.farmaciola.cat.      A      82.32.12.5 ; FQDN ==> no s'expandeix amb $ORIGIN.  
www                      A      82.32.12.2 ; s'expandeix a www.$ORIGIN = www.farmaciola.cat.  
                           A      82.32.12.3 ; blanc ==> node anterior, www en aquest cas  
                           A      82.32.12.4  
-----
```

Fixeu-vos com els dos darrers RRs no tenen propietari (nom de domini). Aquesta és una abreviació habitual: quan un RR no té nom de domini associat, el nom de domini s'assumeix que és el darrer que s'ha especificat. D'aquesta forma, el node **www.farmaciola.cat**. té tres RRs tipus **A**, és a dir, tres adreces. Això permet utilitzar el DNS per balancejar la càrrega entre diferents servidors amb el mateix nom (i diferent adreça) de forma senzilla. La majoria dels servidors DNS implementen un mecanisme de *round-robin* amb què alteren l'ordre amb què els RRs d'un *RR-set* se serveixen. Així, si algú demana per **www.farmaciola.cat**. el primer RR del conjunt retornat serà 82.32.12.2. La segona vegada, 82.32.12.3. La tercera 82.32.12.4. La quarta, 82.32.12.2, i així successivament.

2.22 CONFIGURACIÓ DE SERVIDORS DNS

El servidor DNS que configurarem al laboratori és el BIND (*Berkeley Internet Name Server*), mantingut actualment per l' *Internet Systems Consortium* (ISC). Aproximadament el 80% dels servidors DNS del món es basen en el BIND. Un servidor BIND requereix tres inputs per funcionar:

- **Fitxers de zona**, en el format estàndard comentat, per les zones sobre les que té autoritat; excepte si el servidor és *caching-only* o només fa d'esclau.
- Un fitxer de amb la llista de noms i adreces dels servidors arrel.
- Un fitxer de **configuració** general (habitualment anomenat `named.conf`), on s'indica, entre d'altres:
 - Quines zones hi ha i la ubicació dels fitxers de zona
 - Si el servidor és *master* o *slave* i de/per quines zones.
 - Si permet recursió o no.
 - Control d'accés per: peticions, actualitzacions (en cas de DNS dinàmic) i *zone transfers*.

A continuació teniu un fragment d'exemple de fitxer de configuració, on es comenten les línies més importants. El nom amb què es declaren les zones (p.ex. `zone "farmaciola.cat"`) pot ser qualsevol. Tot i això, és important recordar que, si no s'especifica explícitament, la variable `$ORIGIN` pren el nom de la zona.

Segons la versió del BIND, la configuració pot trobar-se repartida en més d'un fitxer.

<pre> options { directory "/etc/namedb"; allow-query { any; }; recursion no; }; zone "farmaciola.cat" { type master; file "farmaciola.cat.db"; allow-update {none}; allow-transfer { 192.168.1.14; 192.168.3.23; }; }; zone "32.82.in-addr.arpa" { type master; file "reverse-farmaciola.db"; allow-transfer{ 192.168.1.14; 192.168.3.23; }; }; zone "barretina.cat" { type slave; file "barretina.cat.bk"; masters { 192.168.4.12; }; }; zone "."{ type hint; file "root.servers"; }; </pre>	<ul style="list-style-type: none"> ➢ Directori principal ➢ El servidor acceptarà peticions des de qualsevol adreça IP ➢ ...però denegarà peticions recursives <ul style="list-style-type: none"> ➢ Declaració d'una zona ➢ el servidor serà master per la zona ➢ especificuem el nom del fitxer de zona ➢ no permetem actualitzacions de la zona (amb DNS dinàmic) ➢ només es poden fer <i>zone transfer</i> des d'aquestes IPs <ul style="list-style-type: none"> ➢ Declaració de la "zona inversa" ➢ el servidor serà master ➢ especificuem el nom del fitxer de zona ➢ només es poden fer <i>zone transfer</i> des d'aquestes IPs <ul style="list-style-type: none"> ➢ Declaració d'una altra zona ➢ el servidor serà slave per aquesta zona ➢ especificuem el nom del fitxer on el slave guardarà una còpia de la zona ➢ especificuem l'adreça del servidor master de la zona. <ul style="list-style-type: none"> ➢ Fitxer amb la informació dels servidors arrel.
---	---

2.23 EINES DE DIAGNÒSTIC DEL DNS

Com s'ha comentat, les aplicacions (com la web, el correu electrònic etc..) no interactuen directament amb servidors DNS sino que utilitzen un *resolver* per tal de fer peticions DNS. Per aquest motiu, el funcionament del DNS no es perceptible a l'usuari i és transparent a les aplicacions: el procés de resolució s'executa "automàticament". Existeixen però eines per interrogar servidors de noms i interactuar-hi, usades molt sovint com a eines de diagnòstic per verificar el correcte funcionament del sistema o la configuració dels servidors. Les dues eines més populars són el **nslookup** i el **dig**. A continuació teniu un breu resum d'algunes de les opcions que suporta cada eina. Per més informació, podeu utilitzar la comanda `man`.

2.23.1 NSLOOKUP

El nslookup pot funcionar en mode interactiu i no-interactiu. Si s'invoca sense cap paràmetre, arrenca en mode interactiu, permetent a l'usuari formular vàries peticions. A continuació s'explica només el mode interactiu. De fet, el nslookup no el farem servir al laboratori i només s'explica perquè el S.O. Windows no incorpora el dig.

Quan arrenca en mode interactiu, el programa informa sobre el servidor predeterminat a qui enviarà les peticions. L'adreça d'aquest servidor se sol configurar manualment al PC. Per especificar un servidor diferent es pot usar la comana (`server IP`). Les versions més modernes del nslookup detecten automàticament si s'està intentant resoldre una adreça (és a dir trobar el nom o RR tipus PTR associat a una IP) o un nom (és a dir trobar RRs tipus A). Per especificar que es vol consultar un tipus de RR en particular, cal fer `set q=tipus`, on el darrer indica el tipus de RR (com A, MX, PTR, NS, ANY, TXT...etc.). A partir d'aquell moment, l'eina demana els RRs d'aquell tipus per cada nom que introduïu. A continuació teniu un exemple d'una sessió interactiva amb el nslookup.

```
$ nslookup
Servidor predeterminado: backus.upc.es
Address: 147.83.2.3

> www.avui.cat
Servidor: backus.upc.es
Address: 147.83.2.3

Respuesta no autoritativa:
Nombre: arrelli.avui.cat
Address: 81.25.117.130
Aliases: www.avui.cat

> set q=MX
> avui.cat
Servidor: backus.upc.es
Address: 147.83.2.3

Respuesta no autoritativa:
avui.cat      MX preference = 20, mail exchanger = relay.elpunt.net
avui.cat      MX preference = 10, mail exchanger = mail.elpunt.net

avui.cat      nameserver = dns.avui.cat
avui.cat      nameserver = dns147.avui.cat
avui.cat      nameserver = dns148.avui.cat
dns.avui.cat  internet address = 81.25.117.147
dns147.avui.cat internet address = 81.25.117.147
dns148.avui.cat internet address = 81.25.117.148
> exit
```

2.23.2 DIG

El **dig** (*domain information groper*) és més modern que el nslookup, proporciona més informació i és considerat més potent. No suporta un mode interactiu, però pot processar peticions en *batch mode* llegint d'un fitxer o script. De forma simplificada, el format més habitual d'invocar el programa és la següent:

```
dig @server [-c qclass] [-t qtype] domain_name [opcions]
```

on `server` és l'adreça IP del servidor a qui es vol adreçar la petició, `qclass` és la classe que es vol interrogar, `qtype` indica el tipus de RR que es vol demanar i `domain_name` és el nom de domini que es vol consultar. La classe per defecte és `IN`. El tipus per defecte és `A`. L'eina suporta moltes opcions (per consultar-les: `man dig` i/o `dig -h`).

La figura següent mostra un exemple de la informació que proporciona el programa. Fixeu-vos en el tipus de petició, l'adreça del servidor que s'ha consultat i en el format de la sortida del programa.

```

API@ubuntu:~> dig @147.83.2.3 -t A www.elpunt.cat

; <>> DiG 9.7.1 <>> @147.83.2.3 -t A www.elpunt.cat
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31166
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 0

;; QUESTION SECTION:
;www.elpunt.cat.           IN      A

;; ANSWER SECTION:
www.elpunt.cat.          65015   IN      CNAME   arrel.elpunt.cat.
arrel.elpunt.cat.         2183    IN      A       81.25.117.130

;; AUTHORITY SECTION:
elpunt.cat.               42146   IN      NS      dns.elpunt.cat.
elpunt.cat.               42146   IN      NS      dns147.elpunt.cat.
elpunt.cat.               42146   IN      NS      dns148.elpunt.cat.

;; Query time: 1 msec
;; SERVER: 147.83.2.3#53(147.83.2.3)
;; WHEN: Mon Mar 14 17:54:21 2011
;; MSG SIZE rcvd: 128

API@ubuntu:~>

```

2.23.3 RNDC

El **rndc** (*remote name daemon control*) és una eina que incorpora el servidor BIND que permet gestionar-lo de forma senzilla remotament. La configuració del BIND permet especificar si el servidor serà gestionable via rndc i amb quins paràmetres de configuració (p.ex. claus, per motius de seguretat). La forma d'invocar el rndc és:

```
rndc [options] [command] [command-options]
```

Algunes de les opcions més utilitzades són les següents:

-s server	: Adreça IP del servidor a qui es vol enviar la comanda/ordre.
reload <zona>	: Recarrega els fitxers de configuració i de zona. Si s'especifica un nom de zona, carrega aquella zona únicament.
reconfig	: recarrega els fitxers de configuració però només les zones que són noves.
retransfer <zona>	: Torna a transferir una zona sense considerar el serial
stop	: Atura el servidor. Existeix també la opció halt.
status	: Mostra l'estat del servidor.
flush	: Esborra la caché del servidor.

2.24 EL DNS A INTERNET

Tal i com passa amb l'espai d'adreses IPv4, l'espai de noms a Internet està regulat per una organització anomenada ICANN (*Internet Corporation for Assigned Names and Numbers*). L'ICANN és una organització sense ànim de lucre que opera a nivell internacional i que s'encarrega de l'assignació d'adreses IP, d'identificadors de protocol, de la gestió del sistema de noms de domini de primer nivell (gTLDs i ccTLDs), i de la coordinació dels servidors arrel. Aquesta darrera tasca la porta a terme una de les seves branques, l'anomenat RSSAC (DNS Root Server Advisory Committee), mentre que les anteriors les duu a terme el IANA (*Internet Assigned Numbers Authority*). Pel que fa als noms de domini, l'ICANN/IANA s'encarrega de supervisar la delegació de noms de domini de primer nivell (p.ex. .com, .info., etc.).

Els *top-level domains* (TLDs) són els dominis al punt més alt de la jerarquia de l'espai de noms del DNS. L'ICANN delega la gestió d'aquests dominis a entitats acreditades. Històricament els TLDs s'han classificat en categories segons el propòsit del TLD o qui l'ha gestionat. Actualment, algunes de les categories que el IANA distingeix són:

1. Per una banda, els *country-code TLDs* (o **ccTLDs**) estan associats a països. El nom associat a aquests dominis segueix, en la majoria de casos, la recomanació ISO-3166 per identificar cada país amb 2 caràcters (p.ex. **.es** per Espanya, **.it** per Itàlia, **.gr** per Grècia, etc.).
2. D'altra banda hi ha els generic TLDs (**gTLDs**) la denominació dels quals sol permetre identificar el tipus d'organització dels seus subdominis. Dins dels gTLDs, se sol distingir entre els *unsponsored TLDs (uTLDs)* i els *sponsored TLDs (sTLDs)*. Els uTLDs són gestionats sota les polítiques de l'ICANN. Exemples d'aquests TLDs són el **.com**, **.net**, **.info**, o **.org**. Els sTLDs estan esponsoritzats per agències privades que en determinen les polítiques de ús. Per exemple, el domini **.cat** és un gTLD gestionat per la fundació puntCAT. El requeriment que s'exigeix a una organització per tenir un subdomini d'aquest domini és, entre d'altres l'ús de la llengua catalana (http://www.puntcat.cat/normativa/normativa_registro.html). Altres sTLDs són el **.edu**, **.int** o el **.jobs**.
3. *Infrastructure TLDs*. Bàsicament l'únic TLD que pertany a aquesta categoria és el **.arpa** que permet fer la resolució inversa de noms, gràcies als subdominis **in-addr** (per IPv4) i **ip6** (per IPv6, com veureu en propers capítols). També inclou el subdomini **e164** per poder resoldre números de telèfon en el context de veu-sobre-IP (VoIP).

2.24.1 REGISTRIES, REGISTRARS I REGISTRANTS

Un registre de noms de domini (*domain name registry*) és una base de dades que conté informació sobre els dominis registrats a un cert TLD. Els *registries* s'encarreguen de gestionar l'espai de noms de cada TLD, operar el TLD i establir les polítiques d'ús i reserva de noms a un cert TLD. El IANA és un *registry* que delega la gestió de certs TLDs a altres *registries*. D'altra banda, un *Registrar* és una organització (o empresa) acreditada per un registry d'un cert TLD per gestionar la reserva de noms dins d'un TLD, d'acord amb les regles d'ús dictades pel registry. Els *registrars* són les entitats a qui particulars o empreses (un *registrant*) contacten per sol·licitar la reserva i enregistrament d'un cert nom de domini.

2.25 ROOT SERVERS

Els *root servers* (servidors arrel) d'Internet són els servidors que contenen la zona arrel (*root zone*) del DNS. És a dir, són els servidors que serveixen la part de l'espai de noms al punt més alt de la jerarquia de noms. Els *root servers* són els servidors a qui un servidor DNS qualsevol interroga quan se li demana per un nom de domini sobre el que no té cap informació. La zona arrel que mantenen aquests servidors conté informació de delegació pels top-level domains (TLDs). És a dir, RRs tipus **NS** i tipus **A** amb els noms i adreces dels servidors master dels TLDs.

Per limitacions històriques del protocol (bàsicament el fet que l'especificació original limitava la mida màxima d'un missatge UDP a 512 bytes; restricció que, amb d'altres, es va eliminar amb l'anomenat EDNS0, *Extensions mechanisms for DNS*), el nombre de servidors arrel està limitat a 13. La denominació d'aquests servidors és **x.root-server.net** on **x** és **A,B,C,...,M**.

Tot i això no existeixen 13 servidors arrel sinó que molts més. Per motius d'eficiència i robustesa, cada servidor està replicat i existeixen varíes instàncies de cada servidor repartides pel món. A l'any 2011, hi ha 241 servidors arrel, de manera que cadascun dels 13 servidors arrel lògics és en realitat un *cluster* de varis servidors. A més, no hi ha una única entitat que manegui tots els servidors. Aquests estan gestionats o bé per l'ICANN directament o bé per grans empreses (com Verisign), organitzacions com l'*Internet Systems Consortium (ISC)*, o fins i tot la NASA. La figura següent mostra la ubicació d'alguns d'aquests servidors. Per més informació podeu consultar la pàgina web www.root-servers.org.



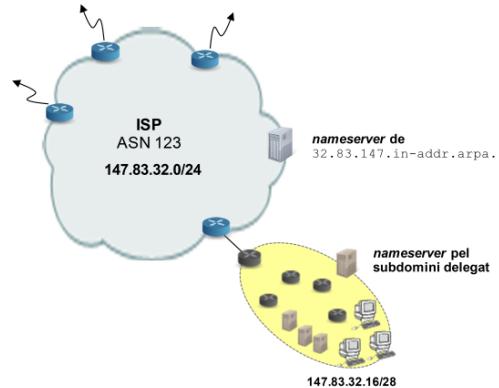
Com s'accedeix a aquests servidors? Doncs la tendència és cada cop més fer servir encaminament *anycast*. És a dir, fer que les instàncies dels servidors “escoltin” (i anuncien) la mateixa adreça IP de manera que les peticions d’altres servidors DNS s’encaminin a la instància més propera (en termes d’encaminament). Per exemple, l’*Internet Systems Consortium* manté 3 servidors arrel F a Europa Occidental, a **Barcelona**, **Munich** (Alemanya) i **Podgorica** (Montenegro). Tots ells fan servir l’adreça **anycast 192.5.5.241**, que s’adverteix via BGP amb ASN 3557, com es mostra la figura.



2.26 DELEGACIÓ DE SUBDOMINIS A IN-ADDR.ARPA. SENSE CLASSES

A mesura que es va adoptar el CIDR, va començar a ser habitual l’assignació de rangs d’adreces IP diferents de /8, /16 o /24 a les diferents organitzacions que formaven Internet. Això va suposar un problema de cara a la resolució de noms amb el mecanisme explicat a la secció 1.10, que, com s’ha vist, només permet delegar zones de `in-addr.arpa.` per rangs /8, /16 o /24. El problema s’il·lustra amb el següent exemple.

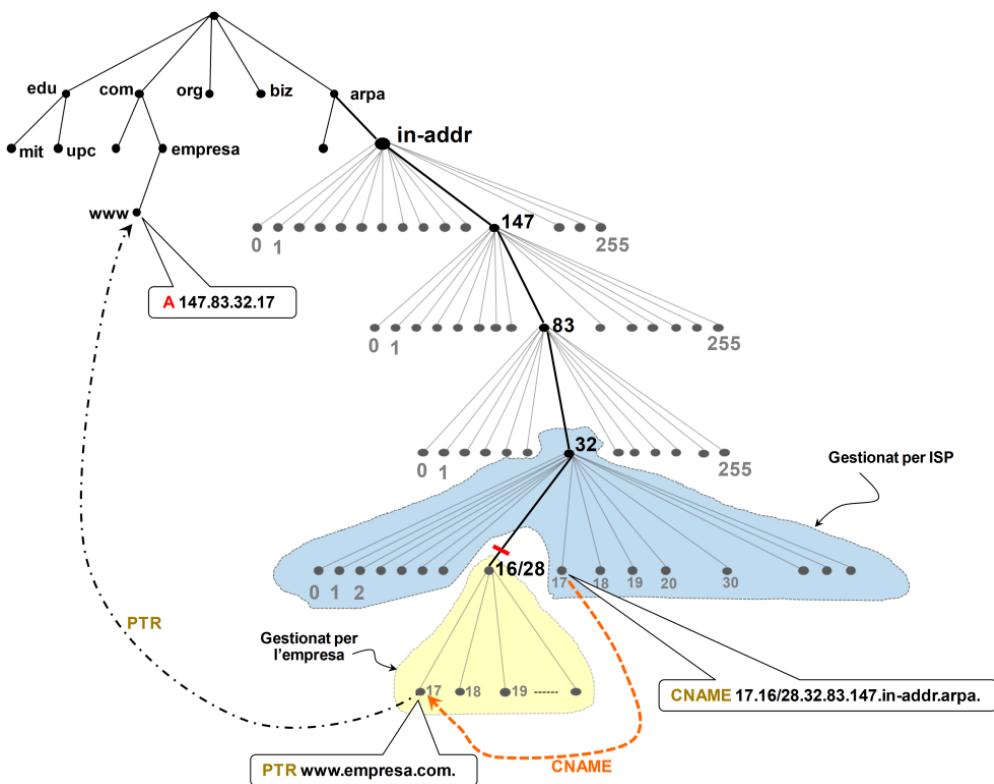
Suposeu un proveïdor d'accés a Internet (ISP) que té assignat el rang **147.83.32.0/24**. Des de la perspectiva del DNS, aquest ISP podria tenir l'autoritat sobre tot el domini `32.83.147.in-addr.arpa`. Aquest ISP podria cedir rangs dins del seu /24 esmentat a xarxes clients. Suposeu que cedís el rang **147.83.32.16/28** (és a dir les adreces .16, .17, .18, .19 ...31) a una certa empresa (amb domini `empresa.com`) a la qual oferís accés a Internet (des d'un punt de vista d'encaminament, ara que coneixeu el BGP, l'ISP advertiria tot el seu /24 --que inclou el /28 del client-- a sistemes autònoms veïns). Des del punt de vista del DNS, però, a l'ISP podria no interessar-li haver de gestionar la resolució d'adreces del rang /28 del la xarxa d'aquest client. I, d'altra banda, és molt possible que el client volgués gestionar ell mateix la resolució d'adreces sense haver de dependre del seu proveïdor.



El problema és que, amb el mecanisme clàssic, només hi ha 3 nivells on es pugui fer la delegació. Una possible solució és la que es descriu a l'RFC 2317. Bèsicament la idea consisteix en "afegir" nodes a l'arbre de l'espai de noms i fer servir RRs tipus CNAME. Segons aquesta solució, el problema a l'exemple se solucionaria de la següent forma. Pel rang 147.83.32.16/28 del client, s'afegiria un nou node amb etiqueta "16/28" (és a dir l'adreça de xarxa i la longitud del prefix). D'aquesta manera l'ISP pot delegar tot el sub-arbre que pengi d'aquest nou node (és a dir, el subdomini `16/28.32.83.147.in-addr.arpa.`) i l'empresa client simplement ha de tenir un fitxer de zona per aquest domini (`16/28.32.83.147.in-addr.arpa.`) especificant RRs tipus PTR per cada adreça IP. Finalment l'altra cosa que hauria de fer el servidor de noms de l'ISP és afegir àlies per cadascuna de les adreces del rang /28 del client.

Així, si al servidor de noms de l'ISP li arriba, per exemple, una petició per resoldre l'adreça 147.83.32.17 (el PTR de `17.32.83.147.in-addr.arpa.`), aquest veuria que el corresponent nom canònic és `17.16/28.32.83.147.in-addr.arpa.` (l'autoritat del qual la tindria el client), i preguntaria al servidor del client (o faria un *referral*) pel corresponent RR tipus PTR.

Amb aquesta solució, el client podria triar els noms que volgués a `16/28.32.83.147.in-addr.arpa.`, sense la intervenció de l'ISP.



ACTIVITATS AL LABORATORI

Objectius de la pràctica

- Entendre el funcionament del protocol DNS.
- Entendre l'estructura del sistema DNS.
- Aprendre a configurar el servidor DNS màster d'una zona.
- Aprendre a configurar un servidor DNS esclau d'una zona.
- Aprendre a delegar un subdomini.

La pràctica està pensada per treballar en un únic PC, que a partir d'ara s'anomenarà **PC**. La figura de l'escenari de la pràctica està al final de tot d'aquest enunciat.

Escenari P08-E01

Descarregueu el fitxer `P08.zip` que conté els fitxers de la pràctica 8 de l'Atenea i **guardieu-lo a l'escriptori** (no canvieu el nom del fitxer).

Obriu un terminal del **PC** i executeu la comanda: `unzip-files P08`

(Si a l'executar `l'unzip-files us` pregunta si voleu substituir (*replace*) algun fitxer, contesteu: `A + Enter`)

A continuació arrancareu els scripts per configurar la topologia de l'escenari, les adreces IPv4 de les interfícies dels routers i dels PCs i l'encaminament.

Executeu al terminal del **PC** la comanda: `P08-E01-start`

Triga una mica. Espereu a que acabi d'executar-se l'script i surti el prompt del terminal.

Executeu al terminal del **PC** la comanda: `P08-DNS-config`

Exercici 1. Anàlisi del protocol DNS

Acobleu un terminal al PC03 de l'escenari amb la comanda: `lxc-attach -n PC03`

Tot aquest exercici l'heu de fer des d'aquest terminal del PC03.

Comproveu el contingut del fitxer `/etc/resolv.conf` del PC03 executant la comanda: `more /etc/resolv.conf`

```
root@api-mv:~# more /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#       DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 147.83.0.1
```

Al fitxer `/etc/resolv.conf` hi ha la llista de servidors de noms a qui el *resolver* (és a dir, el procés que s'encarrega de fer les peticions DNS) fa les peticions DNS quan necessita algun RR. En aquest cas, hi ha l'adreça IP d'un servidor de noms de la UPC.

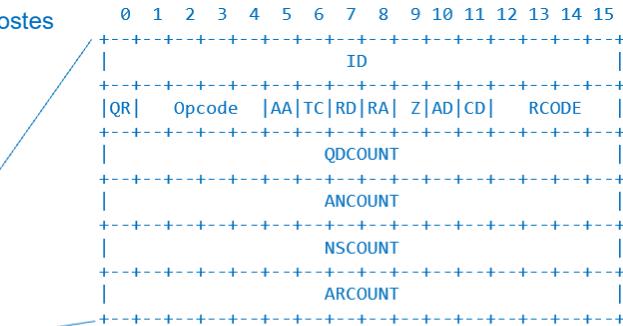
En aquesta pràctica, se us demanarà que editeu i modifiqueu el contingut del fitxer `/etc/resolv.conf` per afegir o treure servidors de noms als que els contenidors de l'escenari fan les peticions DNS. Cal que tingueu present que en les distribucions actuals de Linux (i Ubuntu en particular) el fitxer `/etc/resolv.conf` és, de fet, un enllaç simbòlic que apunta al fitxer `/run/resolvconf/resolv.conf`. Per tal d'afegir i treure servidors de noms de forma permanent no s'hauria d'editar directament el fitxer `/etc/resolv.conf`. perquè el seu contingut no és permanent. Una possible manera (no és l'única) de modificar la llista de servidors de noms consisteix en editar el fitxer `/etc/resolvconf/resolv.conf.d/head`, afegir-hi o treure els servidors de noms que es vulguin i, després, iniciar (o reiniciar si ja està engegat) el servei `resolvconf.service` amb la comanda `systemctl start(restart) resolvconf.service`. Als PCs del laboratori 235G i a la imatge de la màquina virtual d'API o de l'Escriptori virtual, el fitxer `/etc/resolv.conf` que té cada contingut LXC de l'escenari, no és un enllaç simbòlic, per això el podeu editar directament per modificar la llista de servidors DNS. Simplement tingueu present que no és el mecanisme que, en general, s'hauria de fer servir en les noves distribucions per afegir o treure servidors de noms a la llista.

Des del terminal del PC03, utilitzeu l'eina `dig` per realitzar les consultes DNS que s'indiquen i, per cadascuna de les consultes, respondeu les preguntes següents:

- Quin servidor respon? Quins *flags* hi ha a la capçalera de la resposta? Què signifiquen?
- Comenteu el contingut de les seccions QUESTION i ANSWER, és a dir, comproveu quin RR esteu demanant i quina resposta us retorna el servidor. (Si no hi ha resposta, fixeu-vos en el codi d'error)
- Hi ha altres seccions? (Si n'hi ha, comenteu-ne el contingut)

- `dig -t A www.upc.es`
- `dig -t A www.upc.edu`
- `dig @147.83.0.1 -t A www.upc.edu`
- `dig -t PTR 135.2.83.147.in-addr.arpa`
- `dig -t MX upc.edu`
- `dig -t A www.google.es`
- `dig @216.239.36.10 -t A www.google.es`
- `dig @8.8.8.8 -t A www.google.es`
- `dig @216.239.36.10 -t A www.upc.es`
- `dig @8.8.8.8 -t A www.upc.es`
- `dig -t A test.upc.es`
- `dig -t ANY upc.es`

Per entendre el contingut de les peticions i les respostes DNS, cal tenir present el format dels missatges:



QTYPE = tipus de RR
 QCLASS = classe de RR
 QNAME = nom de domini

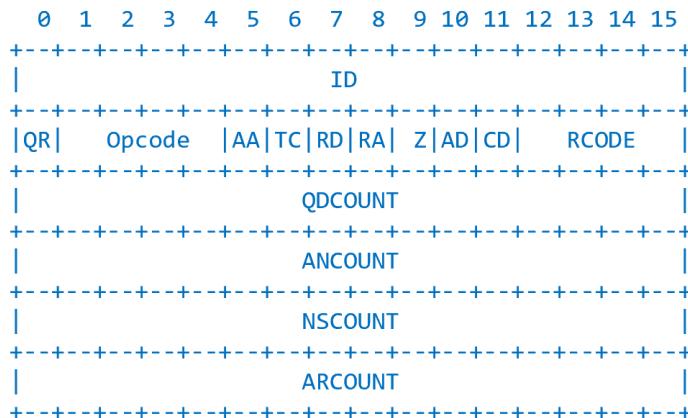
RRs de la resposta

RRs indicant l'autoritat sobre el nom
preguntat (NS).

Opcionalment el SOA.

RRs amb informació relacionada amb la pregunta
(per exemple, tipus A dels NS de la secció autoritat)

El significat dels camps de la capçalera és el següent:



ID → Identificador de 16 bits que assigna el programa que genera la petició. El codi es copia a la resposta i serveix per enllaçar peticions i respostes.

QR → **Query / Response**. QR=0 a les peticions i QR=1 a les respostes.

Opcode → Camp de 4 bits que especifica el tipus de pregunta del missatge. El fixa qui fa la pregunta i es copia a la resposta. Els valors que pot tenir són:

- 0 (QUERY) Pregunta estàndard
- 1 (IQUERY) Pregunta inversa (obsolet)
- 2 (STATUS) Petició de l'estat del servidor
- 3-15 Reservats per a ús futur

AA → **Authoritative Answer**. Aquest bit és vàlid a les respostes i indica que el servidor que respon és autoritatius del domini de la pregunta.

TC	→ Truncation. Especifica que el missatge s'ha truncat per tenir una longitud més gran que la permesa pel canal de transmissió.
RD	→ Recursion Desired. S'activa en la pregunta quan qui fa la pregunta sol·licita recursivitat, és a dir, demana al servidor que, si no té la resposta, la busqui fent peticions iteratives fins que l'obtingui.
RA	→ Recursion Available. S'activa en la resposta si el servidor suporta recursivitat.
Z	→ Reservat per usos futurs
AD	→ Authentic Data. En una petició AD=1 indica que el <i>resolver</i> demana al servidor que digui si els RRs de les seccions resposta i autoritat s'han validat segons la política del servidor. En una resposta AD=1 indica que les dades incloses han estat validades i AD=0 indica que algun dels RRs no s'ha validat o no és segur.
CD	→ Checking Disabled. En una petició CD=1 indica que el <i>resolver</i> accepta que el servidor no realitzi la validació via DNSSEC de les respostes.
RCODE	→ Camp de 4 bits que pren valor en les respostes i pot tenir els codis següents:
	0 (NOERR) No hi ha error 1 (FORMERR) Error de format (no s'ha pogut entendre la pregunta) 2 (SERVFAIL) El servidor de noms no pot processar la petició perquè té algun problema 3 (NXDOMAIN) El nom de domini no existeix. 4 (NOTIMPL) El servidor de noms no suporta el tipus de petició que se li realitza 5 (REFUSED) El servidor de noms rebutja realitzar la petició per la política configurada. 6-15 Reservats per us futur.
QDCOUNT	→ Indica el número d'entrades de la secció Question
ANCOUNT	→ Indica el número de RRs de la secció Answer
NSCOUNT	→ Indica el número de RRs de la secció Authority
ARCOUNT	→ Indica el número de RRs de la secció Additional

En les peticions i respostes DNS que fareu durant la pràctica, veureu que a la secció addicional sovint us surt el pseudo-RR tipus OPT. Per entendre el seu significat cal que conegueu l'existència de l'especificació **Extension mechanisms for DNS (EDNS)** definida amb l'objectiu d'ampliar els paràmetres del protocol DNS. El primer conjunt d'extensions, EDNS0, es va publicar al RFC 2671 i el segon conjunt d'extensions, EDNS(0), es va publicar al RFC 6891. Com que no es poden afegir *flags* a la capçalera DNS, l'extensió EDNS afegeix informació als missatges DNS a través de pseudo-RRs que s'inclouen a la secció de dades addicionals dels missatges DNS. Els pseudo-RRs no apareixen als fitxers de zona, només es transmeten als missatges DNS. EDNS defineix un únic pseudo-RR de tipus OPT que permet especificar la longitud del paquet UDP, ampliar el codi de resposta (RCODE), especificar la versió (actualment 0) i proporcionar espai per a 16 *flags* addicionals. A més a més, disposa d'un camp de dades per poder incloure informació en futures versions. Per compatibilitat amb versions anteriors, els servidors de noms només inclouen el pseudo-RR tipus OPT a la resposta si està present a la pregunta.

L'eina `dig` (domain information groper) permet interrogar servidors de noms. Consulteu el manual (`man dig`) per saber totes les opcions que es poden configurar a l'hora de fer les peticions. En aquest primer exercici veureu que les peticions comentades en la resolució s'han fet utilitzant les opcions `+noadflag` (per indicar al servidor que no digui si les seccions resposta i autoritat s'han validat segona la política del servidor) i `+cdflag` (per indicar al servidor que no cal que validi les respostes via DNSSEC). També s'utilitza l'opció `+qr` per veure, per separat, el detall de la petició i de la resposta DNS.

A la captura **Exercici1.pcapng** teniu els paquets que es comenten en aquesta explicació (el número que hi ha entre parèntesis dins el quadre correspon al número de paquet de la captura).

```
1) root@api-mv:~# dig +noadflag +cdflag +qr -t A www.upc.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t A www.upc.es
;; global options: +cmd

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54104
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;;
;; QUESTION SECTION:
;www.upc.es.           IN      A

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54104
;; flags: qr aa rd cd; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 3
;; WARNING: recursion requested but not available
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;;
;; QUESTION SECTION:
;www.upc.es.           IN      A
;;
;; ANSWER SECTION:
www.upc.es.          3600    IN      A      147.83.2.135
;;
;; AUTHORITY SECTION:
upc.es.               172800  IN      NS      chico.rediris.es.
upc.es.               172800  IN      NS      ns1.upc.edu.
upc.es.               172800  IN      NS      sun.rediris.es.
upc.es.               172800  IN      NS      ns2.upc.edu.
;;
;; ADDITIONAL SECTION:
ns1.upc.edu.          172800  IN      A      147.83.0.1
ns2.upc.edu.          172800  IN      A      147.83.0.2

;; Query time: 46 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:41:47 CEST 2020
;; MSG SIZE  rcvd: 176
```

(1)

(2)

En client (el que fa la petició DNS) demana el RR tipus A del nom de domini `www.upc.es`. Com que no s'especifica cap servidor concret, la petició es fa al primer servidor de la llista de *nameservers* que hi ha definits al fitxer `/etc/resolv.conf`, que, en aquest cas, és el servidor DNS amb adreça IP 147.83.0.1. Aquest servidor de noms no té la recursivitat habilitada (flag `ra=0`) i, per tant, si se li demana un RR que no té als seus fitxers de zona, no farà peticions per trobar-lo. En aquest cas, però té la resposta al seu fitxer de zona, és a dir, és autoritatius de la resposta (flag `aa=1`). Fixeu-vos que a la secció AUTHORITY de la resposta apareix una llista amb els servidors de noms autoritatius del domini `upc.es`. I a la secció ADDITIONAL es facilita l'adreça IP d'alguns d'aquests servidors.

L'opció **-t** va seguida del tipus de RR que es vol demanar

```
root@api-mv:~# dig +noadflag +cdflag +qr -t A www.upc.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t A www.upc.es
;; global options: +cmd
```

(1) Petició

(2) Resposta

```
; Query time: 46 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:41:47 CEST 2020
;; MSG SIZE rcvd: 176
```

Aquesta opció està activada per defecte i provoca que s'imprimeixi un comentari inicial com a sortida de la comanda amb les opcions de la petició que s'han aplicat.

Identifica el servidor de noms (i el port de servei del servidor) que proporciona la resposta

(1) Petició

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.5.15	147.83.0.1	DNS	83	Standard query 0xd358 A www.upc.es

> Frame 1: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface any, id 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 10.0.5.15, Dst: 147.83.0.1
> User Datagram Protocol, Src Port: 42566, Dst Port: 53
Domain Name System (query)
 Transaction ID: 0xd358
 Flags: 0x0110 Standard query
 0... = Response: Message is a query
 .000 0.... = Opcode: Standard query (0)
 0. = Truncated: Message is not truncated
 1 = Recursion desired: Do query recursively
 0. = Z: reserved (0)
 1 = Non-authenticated data: Acceptable
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 1
Queries
 > www.upc.es: type A, class IN
Additional records
 > <Root>: type OPT

```
; ->>HEADER<<- opcode: QUERY status: NOERROR, id: 54104
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;www.upc.es. IN A
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
```

Els que surten a la llista de flags són els que tenen valor 1. En aquest cas concret:
rd=1 per indicar que es demana recursivitat
cd=1 per indicar que es permet que no es verifiquin les respostes

El camp ID del paquet està en hexadecimal a la captura del Wireshark (ID=0xd358) i en decimal a la sortida de la comanda dig (ID=54104).

(2) Resposta

No.	Time	Source	Destination	Protoc	Length	Info
2	0.045857295	147.83.0.1	10.0.5.15	DNS	220	Standard query response 0xd358 A www.upc.es A 147.83.2

> Frame 2: 220 bytes on wire (1760 bits), 220 bytes captured (1760 bits) on interface any, id 0
> Linux cooked capture
> Internet Protocol Version 4, Src: 147.83.0.1, Dst: 10.0.5.15
> User Datagram Protocol, Src Port: 53, Dst Port: 42566
▼ Domain Name System (response)
 Transaction ID: 0xd358
 Flags: 0x8510 Standard query response, No error
 1.... = Response: Message is a response
 .000 0.... = Opcode: Standard query (0)
 1.... = Authoritative: Server is an authority for domain
 0. = Truncated: Message is not truncated
 1 = Recursion desired: Do query recursively
 0.... = Recursion available: Server can't do recursive queries
 0.... = Z: reserved (0)
 0. = Answer authenticated: Answer/authority portion was not authenticated by the server
 1 = Non-authenticated data: Acceptable
 0000 = Reply code: No error (0)

Questions: 1
Answer RRs: 1
Authority RRs: 4
Additional RRs: 3
▼ Queries
 > www.upc.es: type A, class IN
▼ Answers
 > www.upc.es: type A, class IN, addr 147.83.2.135
▼ Authoritative nameservers
 > upc.es: type NS, class IN, ns chico.rediris.es
 > upc.es: type NS, class IN, ns ns1.upc.edu
 > upc.es: type NS, class IN, ns sun.rediris.es
 > upc.es: type NS, class IN, ns ns2.upc.edu
▼ Additional records
 > ns1.upc.edu: type A, class IN, addr 147.83.0.1
 > ns2.upc.edu: type A, class IN, addr 147.83.0.2
 > <Root>: type OPT

Els que surten a la llista de flags són els que tenen valor 1.

En aquest cas concret:

qr=1 perquè és una resposta

aa=1 perquè el servidor que respon és autoritatius (té el RR sol·licitat al seu fitxer de zona)

rd=1 perquè qui ha fet la petició ha demanat recursivitat

cd=1 perquè el que fa la petició permet que no es verifiquin les respostes

```
; ; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54104
; ; flags: qr aa rd cd; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 3
; ; WARNING: recursion requested but not available

; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096

; ; QUESTION SECTION:
; www.upc.es.           IN      A

; ; ANSWER SECTION:
www.upc.es.          3600    IN      A      147.83.2.135

; ; AUTHORITY SECTION:
upc.es.              172800  IN      NS      chico.rediris.es.
upc.es.              172800  IN      NS      ns1.upc.edu.
upc.es.              172800  IN      NS      sun.rediris.es.
upc.es.              172800  IN      NS      ns2.upc.edu.

; ; ADDITIONAL SECTION:
ns1.upc.edu.         172800  IN      A      147.83.0.1
ns2.upc.edu.         172800  IN      A      147.83.0.2
```



```

2) root@api-mv:~# dig +noadflag +cdflag +qr -t A www.upc.edu
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t A www.upc.edu
;; global options: +cmd

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39663
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;;
;; QUESTION SECTION:
;www.upc.edu.           IN      A

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39663
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 3
;; WARNING: recursion requested but not available
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;;
;; QUESTION SECTION:
;www.upc.edu.           IN      A

;; ANSWER SECTION:
www.upc.edu.        172800  IN      CNAME   www.upc.es.
www.upc.es.          3600    IN      A       147.83.2.135
;;
;; AUTHORITY SECTION:
upc.es.              172800  IN      NS      ns1.upc.edu.
upc.es.              172800  IN      NS      ns2.upc.edu.
upc.es.              172800  IN      NS      chico.rediris.es.
upc.es.              172800  IN      NS      sun.rediris.es.
;;
;; ADDITIONAL SECTION:
ns1.upc.edu.         172800  IN      A       147.83.0.1
ns2.upc.edu.         172800  IN      A       147.83.0.2

;; Query time: 45 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:42:08 CEST 2020
;; MSG SIZE rcvd: 194

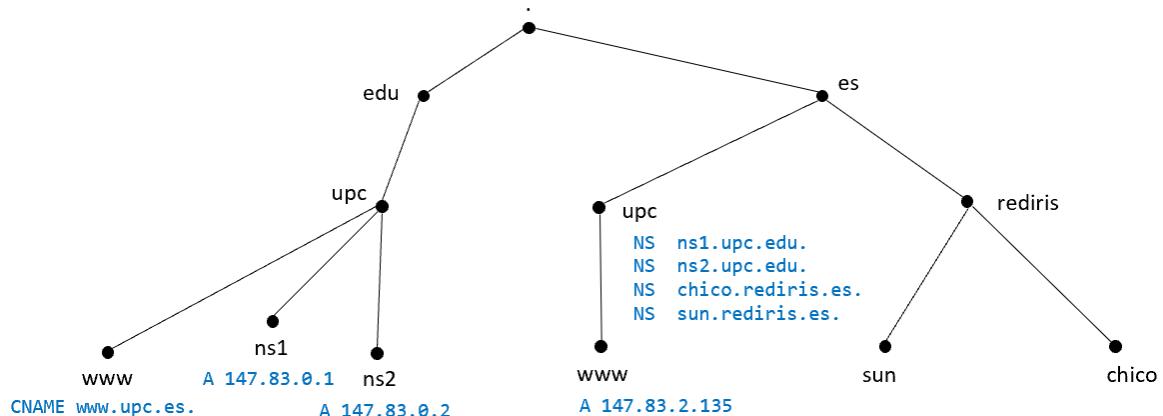
```

(3)

(4)

El client demana el RR tipus A de `www.upc.edu` al servidor de noms amb adreça IP 147.83.0.1. El client demana recursivitat, flag `rd=1` (és a dir, demana que, si el servidor de noms al que es fa la petició no coneix la resposta, faci peticions iteratives fins a trobar-la).

El servidor de noms respon que el vèrtex `www.upc.edu` de l'espai de noms no té un RR tipus A sinó que `www.upc.edu` és un àlies (CNAME) de `www.upc.es`. I que el RR tipus A de `www.upc.es` és 147.83.2.135. El servidor que respon no té la recursivitat habilitada, flag `ra=0`, però pot respondre perquè és autoritatius de la resposta, flag `aa=1` (és a dir, té la resposta al seu fitxer de zona). Tal i com passa a la petició realitzada a l'apartat anterior, el servidor de noms també inclou en el paquet de resposta la llista de servidors de noms autoritatius del domini `upc.es` a la secció AUTHORITY (RRs tipus NS del domini `upc.es`) i algunes de les seves adreces a la secció ADDITIONAL.



```

3) root@api-mv:~# dig +noadflag +cdflag +qr @147.83.0.1 -t A www.upc.edu
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr @147.83.0.1 -t A www.upc.edu
;; global options: +cmd

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64163
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;;
;; QUESTION SECTION:
;www.upc.edu.           IN      A

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64163
;; flags: qr aa rd cd; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 3
;; WARNING: recursion requested but not available
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;;
;; QUESTION SECTION:
;www.upc.edu.           IN      A
;;
;; ANSWER SECTION:
www.upc.edu.        172800  IN      CNAME   www.upc.es.
www.upc.es.          3600    IN      A       147.83.2.135
;;
;; AUTHORITY SECTION:
upc.es.              172800  IN      NS      ns1.upc.edu.
upc.es.              172800  IN      NS      sun.rediris.es.
upc.es.              172800  IN      NS      chico.rediris.es.
upc.es.              172800  IN      NS      ns2.upc.edu.
;;
;; ADDITIONAL SECTION:
ns1.upc.edu.         172800  IN      A       147.83.0.1
ns2.upc.edu.         172800  IN      A       147.83.0.2

;; Query time: 60 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:43:04 CEST 2020
;; MSG SIZE  rcvd: 194

```

(5)

(6)

En aquest cas es fa exactament la mateixa petició que en el cas 2) especificant explícitament al fer la comanda del dig que es vol fer la petició DNS al servidor DNS amb adreça IP 147.83.0.1. Per indicar al dig que es vol interrogar un servidor de noms en particular cal escriure a la comanda dig el símbol @ seguit de l'adreça IP o el nom del servidor que es vol interrogar.

```
4) root@api-mv:~# dig +noadflag +cdflag +qr -t PTR 135.2.83.147.in-addr.arpa
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t PTR 135.2.83.147.in-addr.arpa
;; global options: +cmd
```

```
;; Sending:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11669
; ; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:: udp: 4096
; ; QUESTION SECTION:
; ;135.2.83.147.in-addr.arpa. IN PTR
```

(7)

```
;; Got answer:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11669
; ; flags: qr aa rd cd; QUERY: 1, ANSWER: 11, AUTHORITY: 6, ADDITIONAL: 3
; ; WARNING: recursion requested but not available
```

(8)

```
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:: udp: 4096
; ; QUESTION SECTION:
; ;135.2.83.147.in-addr.arpa. IN PTR

; ; ANSWER SECTION:
135.2.83.147.in-addr.arpa. 172800 IN PTR edicioweb.produccio.upc.edu.
135.2.83.147.in-addr.arpa. 172800 IN PTR www.upc.es.
135.2.83.147.in-addr.arpa. 172800 IN PTR upc.cat.
135.2.83.147.in-addr.arpa. 172800 IN PTR barcelonatech.upc.edu.
135.2.83.147.in-addr.arpa. 172800 IN PTR upc.edu.
135.2.83.147.in-addr.arpa. 172800 IN PTR masters.upc.edu.
135.2.83.147.in-addr.arpa. 172800 IN PTR upc.es.
135.2.83.147.in-addr.arpa. 172800 IN PTR pc6488.upc.es.
135.2.83.147.in-addr.arpa. 172800 IN PTR saladepremsa.upc.edu.
135.2.83.147.in-addr.arpa. 172800 IN PTR cercador.upc.edu.
135.2.83.147.in-addr.arpa. 172800 IN PTR barcelonatech-upc.eu.

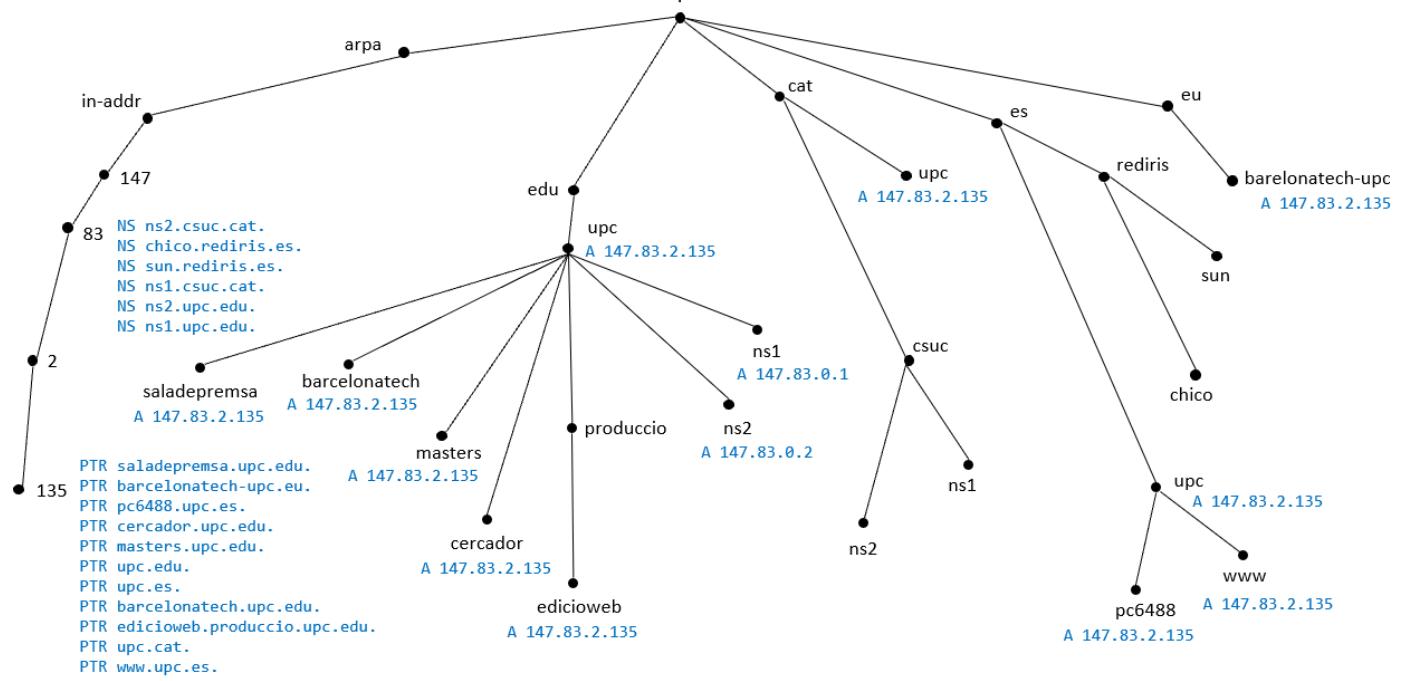
; ; AUTHORITY SECTION:
83.147.in-addr.arpa. 172800 IN NS chico.rediris.es.
83.147.in-addr.arpa. 172800 IN NS ns1.upc.edu.
83.147.in-addr.arpa. 172800 IN NS sun.rediris.es.
83.147.in-addr.arpa. 172800 IN NS ns1.csuc.cat.
83.147.in-addr.arpa. 172800 IN NS ns2.upc.edu.
83.147.in-addr.arpa. 172800 IN NS ns2.csuc.cat.

; ; ADDITIONAL SECTION:
ns1.upc.edu. 172800 IN A 147.83.0.1
ns2.upc.edu. 172800 IN A 147.83.0.2
```

```
; ; Query time: 60 msec
; ; SERVER: 147.83.0.1#53(147.83.0.1)
; ; WHEN: Sun Apr 26 13:43:25 CEST 2020
; ; MSG SIZE rcvd: 478
```

El client demana el RR tipus PTR de 135.2.83.147.in-addr.arpa al servidor de noms amb adreça IP 147.83.0.1. El client demana recursivitat, flag rd=1. L'espai de noms té reservat el domini in-addr.arpa. per a la zona inversa on bàsicament hi ha RRs tipus PTR que permeten saber quins noms s'han assignat a les adreces IP. En el cas de les adreces IPv4, cada nivell de l'espai de noms representa un octet de l'adreça. Per exemple, l'adreça 147.83.2.135 té assignat el vèrtex amb el *Fully Qualified Domain Name* (FQDN) 135.2.83.147.in-addr.arpa. (recordeu que els noms de domini es llegeixen des del vèrtex on hi ha el RR fins a l'arrel).

El servidor de noms respon que l'adreça IP 147.83.2.135 té onze noms diferents. Podeu comprovar, fent la petició corresponent, que a cadascun dels vèrtex apuntats pels RRs tipus PTR del vèrtex 135.2.83.147.in-addr.arpa hi ha un RR tipus A que té com a contingut l'adreça IP 147.83.2.135. El servidor de noms que respon no té la recursivitat habilitada (ra=0) però és autoritatius de la resposta (aa=1), és a dir, la resposta la té en un dels seus fitxers de zona. A la secció AUTHORITY, el servidor de noms inclou la llista de servidors de noms autoritatius de la zona 83.147.in-addr.arpa. i a la secció ADDITIONAL especifica les adreces IPv4 d'alguns d'aquests servidors.



```
5) root@api-mv:~# dig +noadflag +cdflag +qr -t MX upc.edu
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t MX upc.edu
;; global options: +cmd
```

```
; Sending:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55587
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;upc.edu.           IN      MX
```

(9)

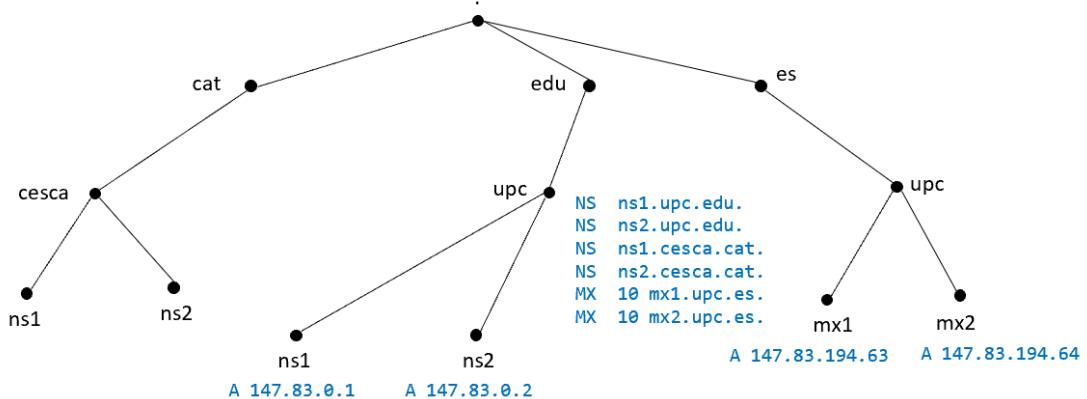
```
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55587
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 5
;; WARNING: recursion requested but not available
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;upc.edu.           IN      MX
;; ANSWER SECTION:
upc.edu.          604800  IN      MX      10 mx1.upc.es.
upc.edu.          604800  IN      MX      10 mx2.upc.es.
;; AUTHORITY SECTION:
upc.edu.          172800  IN      NS      ns2.cesca.cat.
upc.edu.          172800  IN      NS      ns2.upc.edu.
upc.edu.          172800  IN      NS      ns1.upc.edu.
upc.edu.          172800  IN      NS      ns1.cesca.cat.
;; ADDITIONAL SECTION:
mx1.upc.es.       172800  IN      A       147.83.194.63
mx2.upc.es.       172800  IN      A       147.83.194.64
ns1.upc.edu.      172800  IN      A       147.83.0.1
ns2.upc.edu.      172800  IN      A       147.83.0.2
```

(10)

```
; Query time: 47 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:43:43 CEST 2020
;; MSG SIZE  rcvd: 227
```

El client demana els RR tipus MX de `upc.edu` al servidor de noms amb adreça IP 147.83.0.1. El client demana recursitat, flag `rd=1`. Un RR tipus MX té informació sobre un servidor de correu d'un determinat domini. El camp RDATA d'un RR tipus MX és un número (que indica la prioritat assignada al servidor de correu) i un nom (que és el nom del servidor de correu).

El servidor de noms respon que `upc.edu.` té dos RRs tipus MX que són `mx1.upc.edu.` i `mx2.upc.edu..` Tots dos tenen prioritat 10. El servidor de noms que respon no té la recursitat habilitada (`ra=1`) però és autoritatius de la resposta (`aa=1`), és a dir, la resposta la té en un dels seus fitxers de zona. A la secció AUTHORITY, el servidor de noms inclou la llista de servidors de noms autoritatius de la zona `upc.edu.` i a la secció ADDITIONAL especifica algunes adreces d'aquests servidors i també especifica les adreces IP dels servidors de correu.



```
6) root@api-mv:~# dig +noadflag +cdflag +qr -t A www.google.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t A www.google.es
;; global options: +cmd
```

```
; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48960
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.es.           IN      A
```

(11)

```
; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 48960
;; flags: qr rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.es.           IN      A
```

(12)

```
; Query time: 59 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:44:09 CEST 2020
;; MSG SIZE  rcvd: 42
```

El client demana el RR tipus A de `www.google.es` al servidor de noms amb adreça IP 147.83.0.1. El client demana recursitat, flag `rd=1`.

El servidor de noms no té recursitat habilitada (`ra=0`) de manera que no pot fer peticions a altres servidors DNS per buscar la resposta i retorna `status: REFUSED`.

```
7) root@api-mv:~# dig +noadflag +cdflag +qr @216.239.36.10 -t A www.google.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr @216.239.36.10 -t A www.google.es
; (1 server found)
;; global options: +cmd
```

```
;; Sending:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41341
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.es.           IN      A
```

(13)

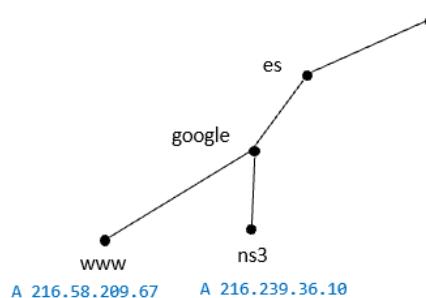
```
;; Got answer:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41341
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.es.           IN      A
;; ANSWER SECTION:
www.google.es.      300     IN      A      216.58.209.67
```

(14)

```
;; Query time: 75 msec
;; SERVER: 216.239.36.10#53(216.239.36.10)
;; WHEN: Sun Apr 26 13:44:35 CEST 2020
;; MSG SIZE rcvd: 58
```

El client torna a demanar el RR tipus A de `www.google.es` però aquesta vegada la petició la fa al servidor de noms amb adreça IP 261.239.36.10. El client demana recursitat (flag `rd=1`)

El servidor de noms al que se li fa la petició és un servidor autoritatius del domini `google.es`. i respon que el RR tipus A de `www.google.es` és 216.58.209.67 (pot ser que us surti una adreça IP diferent). El servidor de noms és autoritatius de la resposta (`ra=1`), és a dir, que té la resposta al seu fitxer de zona. En aquest cas, el servidor de noms no proporciona informació a la secció AUTHORITY ni a la secció ADDITIONAL.



```
8) root@api-mv:~# dig +noadflag +cdflag +qr @8.8.8.8 -t A www.google.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr @8.8.8.8 -t A www.google.es
; (1 server found)
;; global options: +cmd
```

```
;; Sending:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64119
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.es.           IN      A
```

(15)

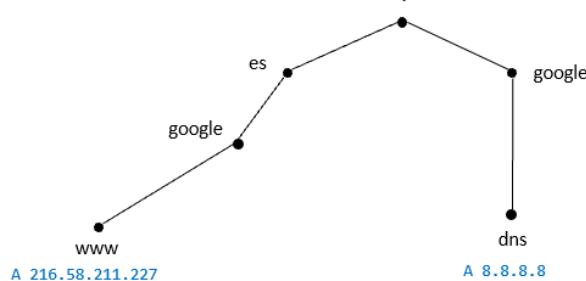
```
;; Got answer:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64119
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.es.           IN      A
;; ANSWER SECTION:
www.google.es.      299     IN      A      216.58.211.227
```

(16)

```
;; Query time: 91 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Sun Apr 26 13:44:53 CEST 2020
;; MSG SIZE  rcvd: 58
```

El client torna a demanar el RR tipus A de `www.google.es` però aquesta vegada la petició la fa al servidor de noms amb adreça IP 8.8.8.8. El client demana recursivitat (flag `rd=1`).

El servidor de noms al que se li fa la petició és un servidor DNS de `google` però no és autoritari del domini `google.es` (no té el fitxer de la zona `google.es`). El servidor té recursivitat habilitada (`ra=1`) i per això pot fer peticions a altres servidors i obtenir la resposta: el RR tipus A de `www.google.es` és 216.58.201.163 (pot ser que us surti una adreça IP diferent). En aquest cas, el servidor de noms no proporciona informació a la secció `AUTHORITY` ni a la secció `ADDITIONAL`.



```
9) root@api-mv:~# dig +noadflag +cdflag +qr @216.239.36.10 -t A www.upc.es
```

```
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr @216.239.36.10 -t A www.upc.es
; (1 server found)
;; global options: +cmd
```

```
;; Sending:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8885
; ; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
; ; QUESTION SECTION:
; www.upc.es.           IN      A
```

(17)

```
;; Got answer:
; ;-->>HEADER<<- opcode: QUERY, status: REFUSED, id: 8885
; ; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
; ; WARNING: recursion requested but not available
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 512
; ; QUESTION SECTION:
; www.upc.es.           IN      A
```

(18)

```
;; Query time: 75 msec
; SERVER: 216.239.36.10#53(216.239.36.10)
; WHEN: Sun Apr 26 13:45:31 CEST 2020
; MSG SIZE rcvd: 39
```

El client demana el RR tipus A de `www.upc.es` al servidor de noms amb adreça IP 261.239.36.10. El client demana recursivitat, flag `rd=1`. El servidor de noms al que se li fa la petició és un servidor autoritatius del domini `google.com`, que no té la recursivitat habilitada i, com que se li demana un RR que no té en el seu fitxer de zona, no pot respondre (status: REFUSED).

```
10) root@api-mv:~# dig +noadflag +cdflag +qr @8.8.8.8 -t A www.upc.es
```

```
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr @8.8.8.8 -t A www.upc.es
; (1 server found)
;; global options: +cmd
```

```
;; Sending:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7345
; ; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
; ; QUESTION SECTION:
; www.upc.es.           IN      A
```

(19)

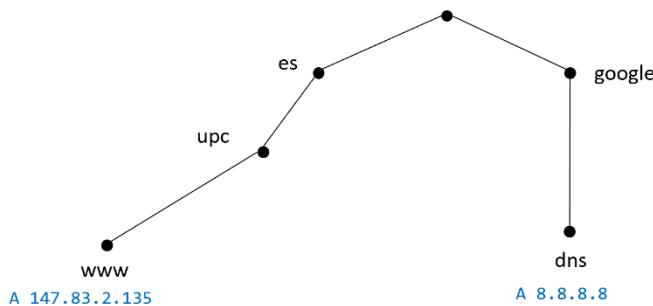
```
;; Got answer:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7345
; ; flags: qr rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 512
; ; QUESTION SECTION:
; www.upc.es.           IN      A
; ; ANSWER SECTION:
www.upc.es.        2329    IN      A      147.83.2.135
```

(20)

```
;; Query time: 83 msec
; SERVER: 8.8.8.8#53(8.8.8.8)
; WHEN: Sun Apr 26 13:45:41 CEST 2020
; MSG SIZE rcvd: 55
```

El client torna a demanar el RR tipus A de `www.upc.es` però aquesta vegada la petició la fa al servidor de noms amb adreça IP 8.8.8.8. El client demana recursivitat, flag `rd=1`. El servidor de noms al que se li fa la petició és un servidor DNS de `google` que sí que té la recursivitat habilitada (`ra=1`) i per això pot fer peticions a altres servidors i obtenir la

resposta: el RR tipus A de `www.upc.es` és 147.83.2.135. En aquest cas, el servidor de noms no proporciona informació a la secció AUTHORITY ni a la secció ADDITIONAL.



```
11) root@api-mv:~# dig +noadflag +cdflag +qr -t A test.upc.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t A test.upc.es
; global options: +cmd
```

```
;; Sending:
; ;-->HEADER<<- opcode: QUERY, status: NOERROR, id: 34851
; ; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
; ; QUESTION SECTION:
; ;test.upc.es.           IN      A
```

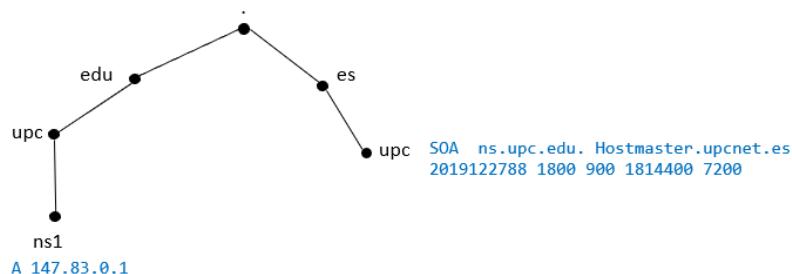
(21)

```
;; Got answer:
; ;-->HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 34851
; ; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
;
; ; QUESTION SECTION:
; ;test.upc.es.           IN      A
;
; ; AUTHORITY SECTION:
upc.es.          7200   IN      SOA     ns.upc.edu. hostmaster.upcnet.es. 2019122788
1800 900 1814400 7200
```

(22)

```
;; Query time: 60 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:45:57 CEST 2020
;; MSG SIZE  rcvd: 104
```

El client demana el RR tipus A de `test.upc.es` al servidor de noms amb adreça IP 147.0.83.1. El client demana recursitat, flag `rd=1`. El servidor de noms al que se li fa la petició no té la recursitat habilitada (`ra=0`) però no li hauria de fer falta buscar la resposta perquè és un servidor autoritatius del domini `upc.es` (`aa=1`). No obstant això, fixeu-vos que no hi ha resposta perquè el RR pel que s'està demandant fa referència a un punt de l'espai de noms que no existeix, per això en el paquet que envia el servidor de noms s'indica `status: NXDOMAIN` i a la secció AUTHORITY hi ha el RR tipus SOA de `upc.es` (que és la zona on hi hauria d'haver el RR pel qual es demana):



```
12) root@api-mv:~# dig +noadflag +cdflag +qr -t ANY upc.es
; <>> DiG 9.10.3-P4-Ubuntu <>> +noadflag +cdflag +qr -t ANY upc.es
;; global options: +cmd
```

```
;; Sending:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61464
; ; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
;
; ; QUESTION SECTION:
;upc.es.          IN      ANY
```

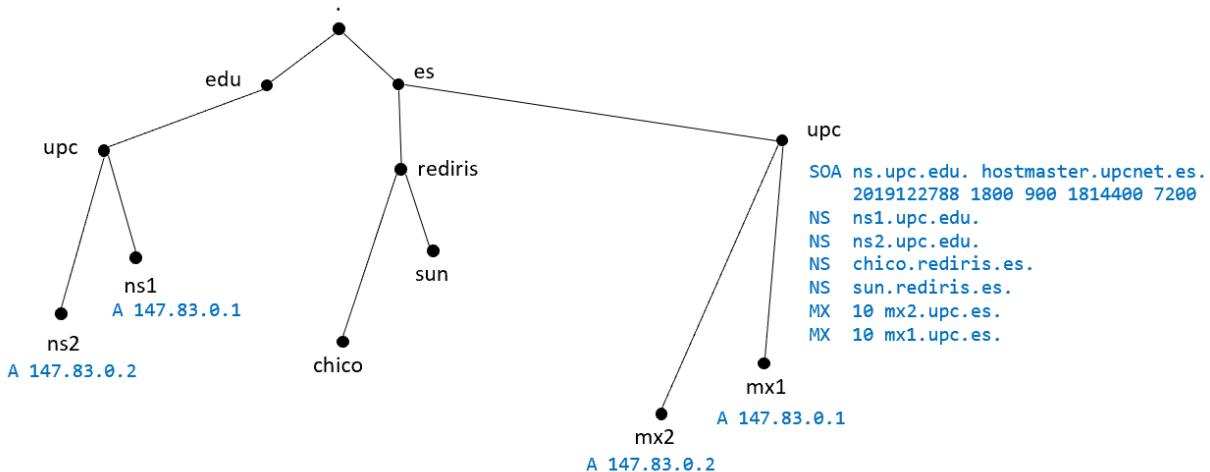
(23)

```
;; Got answer:
; ;-->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61464
; ; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 5
; ; WARNING: recursion requested but not available
;
; ; OPT PSEUDOSECTION:
; ; EDNS: version: 0, flags:; udp: 4096
;
; ; QUESTION SECTION:
;upc.es.          IN      ANY
;
; ; ANSWER SECTION:
upc.es.          172800  IN      MX      10 mx2.upc.es.
upc.es.          172800  IN      MX      10 mx1.upc.es.
upc.es.          172800  IN      NS      ns2.upc.edu.
upc.es.          172800  IN      NS      chico.rediris.es.
upc.es.          172800  IN      NS      ns1.upc.edu.
upc.es.          172800  IN      NS      sun.rediris.es.
upc.es.          172800  IN      SOA     ns.upc.edu. hostmaster.upcnet.es. 2019122788
                                         1800 900 1814400 7200
;
; ; ADDITIONAL SECTION:
mx1.upc.es.      172800  IN      A       147.83.194.63
mx2.upc.es.      172800  IN      A       147.83.194.64
ns1.upc.edu.     172800  IN      A       147.83.0.1
ns2.upc.edu.     172800  IN      A       147.83.0.2
```

(24)

```
;; Query time: 61 msec
;; SERVER: 147.83.0.1#53(147.83.0.1)
;; WHEN: Sun Apr 26 13:46:13 CEST 2020
;; MSG SIZE  rcvd: 285
```

El client demana els RR que hi hagi (ANY) al vèrtex `upc.es` al servidor de noms amb adreça IP 147.0.83.1. El client demana recursivitat, flag `rd=1`. El servidor de noms al que se li fa la petició no té la recursivitat habilitada (`ra=0`) però no li cal fer la pregunta a cap altre servidor perquè és autoritatius del domini `upc.es` (`aa=1`). El servidor respon amb tots els RR que hi ha a `upc.es`. No hi ha secció `AUTHORITY`, però a la secció `ADDITIONAL` el servidor especifica les adreces IP dels servidors DNS i dels servidors de correu de `upc.es`.



Podeu aprofitar el terminal que teniu obert al PC03 per fer l'exercici següent.

Exercici 2. Configuració d'un servidor caché (caching-only nameserver)

A continuació configurareu un servidor DNS al PC03 de l'escenari que, de moment, actuarà només de servidor caché (no tindrà cap fitxer de zona). Si no teniu obert el terminal de l'exercici anterior, acobieu un terminal al PC03 amb la comanda: `lxc-attach -n PC03`

Activeu el *wireshark* per capturar els paquets al bridge br01.

Al terminal del PC03, arranqueu el servidor DNS amb la comanda: `systemctl start bind9`

Des del PC01 de l'escenari, feu una petició DNS al servidor DNS del PC03 amb la comanda següent:

```
lxc-attach -n PC01 -- dig @10.0.1.2 -t A www.upc.es
```

- a) Verifiqueu que la resposta obtinguda és l'esperada (fixeu-vos en el camp *flags*, en el contingut de les diferents seccions, etc.) Analitzeu els paquets DNS enviats i rebuts pel servidor com a conseqüència de la petició realitzada. Per cada petició i resposta analitzeu-ne el contingut (fixeu-vos en els flags, en el contingut de les diferents seccions, etc.) Feu un diagrama on aparegui el PC client i tots els servidors DNS que han intervingut en el diàleg i feu un esquema de les peticions i respostes que s'han enviat entre ells.

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t A www.upc.es
; <>> DiG 9.11.3-1ubuntu1.1-Ubuntu <>> +noadflag +cdflag +qr @10.0.1.2 -t A www.upc.es
;; global options: +cmd

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24970
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.upc.es.           IN      A

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24970
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.upc.es.           IN      A
;; ANSWER SECTION:
www.upc.es.        3600    IN      A      147.83.2.135
;; AUTHORITY SECTION:
upc.es.          86400   IN      NS      ns2.upc.edu.
upc.es.          86400   IN      NS      ns1.upc.edu.
upc.es.          86400   IN      NS      chico.rediris.es.
upc.es.          86400   IN      NS      sun.rediris.es.

;; Query time: 1478 msec
;; SERVER: 10.0.1.2#53(10.0.1.2)
;; WHEN: Tue Apr 25 11:03:32 UTC 2020
;; MSG SIZE  rcvd: 172
```

El client demana el RR tipus A de `www.upc.es` al servidor de noms amb adreça IP 10.0.2.1. El client demana recursivitat, flag `rd=1`.

El servidor de noms al que se li fa la petició té la recursivitat habilitada (`ra=1`). El servidor respon que el RR tipus A de `www.upc.es` de 147.83.2.135. El servidor de noms no és autoritatius de la resposta (flag `aa=0`) i això vol dir que ha hagut de buscar la resposta en un altre servidor.

A la captura **Exercici2_apartatA.pcapng** hi ha el detall dels paquets que s'envien des que el client fa la petició al servidor DNS del PC03 fins que en rep la resposta.

Els paquets 1 i 8 de la captura **Exercici2_apartatA.pcapng** corresponen a la petició que fa el client del PC01 i a la resposta que envia el servidor del PC03.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es
4	1.029905763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

> Frame 1: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface br01, id 0
> Ethernet II, Src: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8), Dst: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12)
> Internet Protocol Version 4, Src: 10.0.1.66, Dst: 10.0.1.2
> User Datagram Protocol, Src Port: 60027, Dst Port: 53
▼ Domain Name System (query)
 Transaction ID: 0x618a
 Flags: 0x0110 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 1
▼ Queries
 > www.upc.es: type A, class IN
▼ Additional records
 > <Root>: type OPT
[Response In: 8]

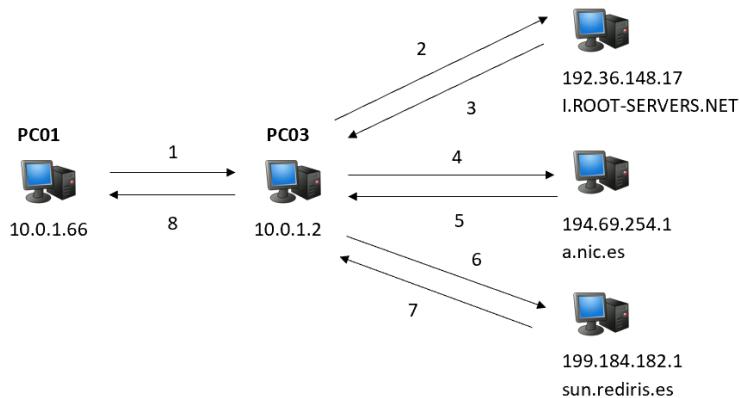
▼ Flags: 0x0110 Standard query
 0... = Response: Message is a query
 .000 0... = Opcode: Standard query (0)
0. = Truncated: Message is not truncated
1 = Recursion desired: Do query recursively
0. = Z: reserved (0)
1 = Non-authenticated data: Acceptable

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es NS ns-es.nic.fi
4	1.029905763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es NS chico.rediris.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

> Frame 8: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface br01, id 0
> Ethernet II, Src: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12), Dst: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8)
> Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.66
> User Datagram Protocol, Src Port: 53, Dst Port: 60027
▼ Domain Name System (response)
 Transaction ID: 0x618a
 Flags: 0x8190 Standard query response, No error
 Questions: 1
 Answer RRs: 1
 Authority RRs: 4
 Additional RRs: 1
▼ Queries
 > www.upc.es: type A, class IN
▼ Answers
 > www.upc.es: type A, class IN, addr 147.83.2.135
▼ Authoritative nameservers
 > upc.es: type NS, class IN, ns ns2.upc.edu
 > upc.es: type NS, class IN, ns chico.rediris.es
 > upc.es: type NS, class IN, ns sun.rediris.es
 > upc.es: type NS, class IN, ns ns1.upc.edu
▼ Additional records
 > <Root>: type OPT
[Request In: 1]

▼ Flags: 0x8190 Standard query response, No error
 1... = Response: Message is a response
 .000 0... = Opcode: Standard query (0)
0. = Authoritative: Server is not an authority for domain
0. = Truncated: Message is not truncated
1 = Recursion desired: Do query recursively
 1.... = Recursion available: Server can do recursive queries
0. = Z: reserved (0)
0. = Answer authenticated: Answer/authority portion was not
1 = Non-authenticated data: Acceptable
 0000 = Reply code: No error (0)

Per poder obtenir la resposta, el servidor 10.0.1.2 (PC03) ha hagut de preguntar pel RR tipus A de `www.upc.es` a altres servidors DNS.



Com que el servidor 10.0.1.2 no té inicialment cap informació a la seva caché, el primer que ha de fer és fer la petició a un servidor arrel, que escull entre tots els servidors arrels de la llista que té guardada en un fitxer. El servidor DNS que s'utilitza a la pràctica, té la llista de servidors arrel guardada al fitxer `db.root` del directori `/etc/bind`. A l'exemple de la captura, el servidor 10.0.1.2 tria el servidor I.ROOT-SERVERS.NET amb adreça IPv4 192.36.148.17 per fer-li la petició del RR tipus A de `www.upc.es` (paquet 2). El servidor 10.0.1.2 no demana recursivitat al servidor arrel (flag `rd=0`) perquè no es desitja que el servidor arrel busqui la resposta pel seu compte si no la té.

El servidor arrel no té la resposta i per això no hi ha secció ANSWER al paquet que envia en resposta a la petició del servidor 10.0.1.2 (paquet 3). No obstant, a la secció AUTHORITY afegeix la llista de servidors autoritatius del domini `.es`. A més a més, a la secció ADDITIONAL inclou les adreces IP d'aquests servidors.

A continuació, el servidor 10.0.1.2 escull un servidor DNS de la llista de servidors autoritatius del domini `.es` i li fa la petició del RR tipus A de `www.upc.es`. A l'exemple de la captura, el servidor 10.0.1.2 tria el servidor `a.nic.es` amb adreça IPv4 194.69.254.1 per fer-li la petició del RR tipus A de `www.upc.es` (paquet 4). El servidor 10.0.1.2 no demana recursivitat (flag `rd=0`) perquè no es desitja que el servidor DNS interrogat busqui la resposta pel seu compte si no la té.

El servidor `a.nic.es` no té la resposta i per això no hi ha secció ANSWER al paquet que envia en resposta a la petició del servidor 10.0.1.2 (paquet 5). No obstant, a la secció AUTHORITY afegeix la llista de servidors autoritatius del domini `upc.es`. A més a més, a la secció ADDITIONAL inclou les adreces IP d'alguns d'aquests servidors. Nota: al paquet 5 es veuen RRs tipus NSEC3 i RRSIG que estan relacionats amb el DNSSEC. El DNSSEC (*Domain Name System Security Extensions*) és un conjunt d'especificacions de l'IETF per tal de verificar la veritat de part de la informació que proporciona el sistema DNS. En aquesta pràctica no ho estudarem.

A continuació, el servidor 10.0.1.2 escull un servidor DNS de la llista de servidors autoritatius del domini `upc.es` i li fa la petició del RR tipus A de `www.upc.es`. A l'exemple de la captura, el servidor 10.0.1.2 tria el servidor `sun.rediris.es` amb adreça IPv4 199.184.182.1 per fer-li la petició del RR tipus A de `www.upc.es` (paquet 6). El servidor 10.0.1.2 no demana recursivitat (flag `rd=0`) perquè no es desitja que el servidor DNS interrogat busqui la resposta pel seu compte si no la té.

El servidor `sun.rediris.es` no només té la resposta si no que és autoritatius de la resposta (la té al seu fitxer de zona). Per això, en la resposta que envia el servidor (paquet 7) el flag autoritatius de la capçalera està activat (flag `aa=1`) i a la secció ANSWER s'especifica que el RR tipus A de `www.upc.es` és 147.83.2.135. El servidor també afegeix la secció AUTHORITY amb la llista de servidors autoritatius del domini `upc.es`.

Quan el servidor 10.0.1.2 té la resposta, ja la pot comunicar al client (el PC amb adreça IP 10.0.1.66). Com que el servidor 10.0.1.2 no té la resposta al seu fitxer de zona, el flag autoritatius està desactivat (flag `aa=0`) en la resposta que envia al client (paquet 8).

A continuació teniu el detall dels paquets (2,3) (4,5) i (6,7) que s'indiquen a la figura i que es mencionen a l'explicació.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es
4	1.029905763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

< >

```

> Frame 2: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface br01, id 0
> Ethernet II, Src: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12), Dst: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8)
> Internet Protocol Version 4, Src: 10.0.1.2, Dst: 192.36.148.17
> Transmission Control Protocol, Src Port: 32991, Dst Port: 53, Seq: 3006368663, Ack: 2506112002, Len: 53
  Domain Name System (query)
    Length: 51
    Transaction ID: 0x25ee
    Flags: 0x0010 Standard query
      Questions: 1
      Answer RRs: 0
      Authority RRs: 0
      Additional RRs: 1
    Queries
      > www.upc.es: type A, class IN
    Additional records
      > <Root>: type OPT
      [Response In: 3]
  
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es NS ns-es.nic.fr
4	1.029905763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es NS chico.redirigir
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

< >

```

> Frame 3: 1006 bytes on wire (8048 bits), 1006 bytes captured (8048 bits) on interface br01, id 0
> Ethernet II, Src: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8), Dst: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12)
> Internet Protocol Version 4, Src: 192.36.148.17, Dst: 10.0.1.2
> Transmission Control Protocol, Src Port: 53, Dst Port: 32991, Seq: 2506112002, Ack: 3006368716, Len: 952
  Domain Name System (response)
    Length: 950
    Transaction ID: 0x25ee
    Flags: 0x8010 Standard query response, No error
      Questions: 1
      Answer RRs: 0
      Authority RRs: 12
      Additional RRs: 15
    Queries
      > www.upc.es: type A, class IN
    Authoritative nameservers
      > es: type NS, class IN, ns ns-es.nic.fr
      > es: type NS, class IN, ns g.nic.es
      > es: type NS, class IN, ns h.nic.es
      > es: type NS, class IN, ns a.nic.es
      > es: type NS, class IN, ns ns1.cesca.es
      > es: type NS, class IN, ns f.nic.es
      > es: type NS, class IN, ns ssdns-tld.nic.cl
      > es: type DS, class IN
      > es: type RRSIG, class IN
    Additional records
      > a.nic.es: type A, class IN, addr 194.69.254.1
      > a.nic.es: type AAAA, class IN, addr 2001:67c:21cc:2000::64:41
      > f.nic.es: type A, class IN, addr 130.206.1.7
      > f.nic.es: type AAAA, class IN, addr 2001:720:418:caf1::7
      > g.nic.es: type A, class IN, addr 204.61.217.1
      > g.nic.es: type AAAA, class IN, addr 2001:500:14:7001:ad::1
      > h.nic.es: type A, class IN, addr 194.0.33.53
      > h.nic.es: type AAAA, class IN, addr 2001:678:40::53
      > ns1.cesca.es: type A, class IN, addr 84.88.0.3
      > ns1.cesca.es: type AAAA, class IN, addr 2001:40b0:1:1122:ce5c:a000:0:3
      > ns-es.nic.fr: type A, class IN, addr 194.0.9.1
      > ns-es.nic.fr: type AAAA, class IN, addr 2001:678:c::1
      > ssdns-tld.nic.cl: type A, class IN, addr 200.7.5.14
      > ssdns-tld.nic.cl: type AAAA, class IN, addr 2001:1398:276:0:200:7:5:14
      > <Root>: type OPT
      [Request In: 2]
  
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es NS ns-es.nic.fr
4	1.029905763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es NS chico.rediris.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

< >

```

> Frame 4: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface br01, id 0
> Ethernet II, Src: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12), Dst: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8)
> Internet Protocol Version 4, Src: 10.0.1.2, Dst: 194.69.254.1
> Transmission Control Protocol, Src Port: 38803, Dst Port: 53, Seq: 1772424213, Ack: 2506240002, Len: 53
`- Domain Name System (query)
  Length: 51
  Transaction ID: 0xed67
  Flags: 0x0010 Standard query
  Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
    ` Queries
      > www.upc.es: type A, class IN
    ` Additional records
      > <Root>: type OPT
      [Response In: 5]

```

< Flags: 0x0010 Standard query
 0... = Response: Message is a query
 .000 0... = Opcode: Standard query (0)
0. = Truncated: Message is not truncated
0 = Recursion desired: Don't do query recursively
0. = Z: reserved (0)
1 = Non-authenticated data: Acceptable

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es NS ns-es.nic.fr
4	1.029905763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es NS chico.rediris.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

< >

```

> Frame 5: 771 bytes on wire (6168 bits), 771 bytes captured (6168 bits) on interface br01, id 0
> Ethernet II, Src: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8), Dst: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12)
> Internet Protocol Version 4, Src: 194.69.254.1, Dst: 10.0.1.2
> Transmission Control Protocol, Src Port: 53, Dst Port: 38803, Seq: 2506240002, Ack: 1772424266, Len: 717
`- Domain Name System (response)
  Length: 717
  Transaction ID: 0xed67
  Flags: 0x8010 Standard query response, No error
  Questions: 1
  Answer RRs: 0
  Authority RRs: 8
  Additional RRs: 5
  ` Queries
    > www.upc.es: type A, class IN
  ` Authoritative nameservers
    > upc.es: type NS, class IN, ns chico.rediris.es
    > upc.es: type NS, class IN, ns ns1.upc.edu
    > upc.es: type NS, class IN, ns ns2.upc.edu
    > upc.es: type NS, class IN, ns sun.rediris.es
    > skb41e8seu2hfj1j2egqf7fq4quesqe7.es: type NSEC3, class IN
    > skb41e8seu2hfj1j2egqf7fq4quesqe7.es: type RRSIG, class IN
    > d832sb13269ouari5ishe2qv1o72m7df.es: type NSEC3, class IN
    > d832sb13269ouari5ishe2qv1o72m7df.es: type RRSIG, class IN
  ` Additional records
    > sun.rediris.es: type A, class IN, addr 199.184.182.1
    > sun.rediris.es: type AAAA, class IN, addr 2620:171:808::1
    > chico.rediris.es: type A, class IN, addr 130.206.1.3
    > chico.rediris.es: type AAAA, class IN, addr 2001:720:418:caf1::3
    > <Root>: type OPT
    [Request In: 4]

```

< Flags: 0x8010 Standard query response, No error
 1... = Response: Message is a response
 .000 0... = Opcode: Standard query (0)
0. = Authoritative: Server is not an authority for domain
0 = Truncated: Message is not truncated
0 = Recursion desired: Don't do query recursively
 0... = Recursion available: Server can't do recursive queries
0. = Z: reserved (0)
0 = Answer authenticated: Answer/authority portion was not
1 = Non-authenticated data: Acceptable
0000 = Reply code: No error (0)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es NS ns-es.nic.fr
4	1.029985763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es NS chico.rediris.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

< >

```

> Frame 6: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface br01, id 0
> Ethernet II, Src: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12), Dst: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8)
> Internet Protocol Version 4, Src: 10.0.1.2, Dst: 199.184.182.1
> User Datagram Protocol, Src Port: 39020, Dst Port: 53
  ↴ Domain Name System (query)
    Transaction ID: 0xc944
    > Flags: 0x0010 Standard query
      Questions: 1
      Answer RRs: 0
      Authority RRs: 0
      Additional RRs: 1
    < Queries
      > www.upc.es: type A, class IN
    < Additional records
      > <Root>: type OPT
      [Response In: 7]
  ↴ Flags: 0x0010 Standard query
    0... .... .... .... = Response: Message is a query
    .000 0.... .... .... = Opcode: Standard query (0)
    .... ..0. .... .... = Truncated: Message is not truncated
    .... ..0. .... .... = Recursion desired: Don't do query recursively
    .... .... 0.. .... = Z: reserved (0)
    .... .... ..1 .... = Non-authenticated data: Acceptable

```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.66	10.0.1.2	DNS	93	Standard query 0x618a A www.upc.es OPT
2	0.313601408	10.0.1.2	192.36.148.17	DNS	107	Standard query 0x25ee A www.upc.es OPT
3	0.475571990	192.36.148.17	10.0.1.2	DNS	1006	Standard query response 0x25ee A www.upc.es NS ns-es.nic.fr
4	1.029985763	10.0.1.2	194.69.254.1	DNS	107	Standard query 0xed67 A www.upc.es OPT
5	1.102444328	194.69.254.1	10.0.1.2	DNS	771	Standard query response 0xed67 A www.upc.es NS chico.rediris.es
6	1.104106747	10.0.1.2	199.184.182.1	DNS	93	Standard query 0xc944 A www.upc.es OPT
7	1.155038068	199.184.182.1	10.0.1.2	DNS	186	Standard query response 0xc944 A www.upc.es A 147.83.2.135
8	1.155428474	10.0.1.2	10.0.1.66	DNS	214	Standard query response 0x618a A www.upc.es A 147.83.2.135

< >

```

> Frame 7: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface br01, id 0
> Ethernet II, Src: e6:4c:84:1e:19:d8 (e6:4c:84:1e:19:d8), Dst: da:83:0c:f7:f8:12 (da:83:0c:f7:f8:12)
> Internet Protocol Version 4, Src: 199.184.182.1, Dst: 10.0.1.2
> User Datagram Protocol, Src Port: 53, Dst Port: 39020
  ↴ Domain Name System (response)
    Transaction ID: 0xc944
    > Flags: 0x8400 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 4
      Additional RRs: 1
    < Queries
      > www.upc.es: type A, class IN
    < Answers
      > www.upc.es: type A, class IN, addr 147.83.2.135
    < Authoritative nameservers
      > upc.es: type NS, class IN, ns ns1.upc.edu
      > upc.es: type NS, class IN, ns ns2.upc.edu
      > upc.es: type NS, class IN, ns sun.rediris.es
      > upc.es: type NS, class IN, ns chico.rediris.es
    < Additional records
      > <Root>: type OPT
      [Request In: 6]
  ↴ Flags: 0x8400 Standard query response, No error
    1... .... .... .... = Response: Message is a response
    .000 0.... .... .... = Opcode: Standard query (0)
    .... ..1. .... .... = Authoritative: Server is an authority for domain
    .... ..0. .... .... = Truncated: Message is not truncated
    .... ..0. .... .... = Recursion desired: Don't do query recursively
    .... .... 0.. .... = Recursion available: Server can't do recursive queries
    .... .... 0.. .... = Z: reserved (0)
    .... .... ..0. .... = Answer authenticated: Answer/authority portion was not
    .... .... ..0. .... = Non-authenticated data: Unacceptable
    .... .... ..0000 = Reply code: No error (0)

```

Reinicieu la captura del *wireshark*. Torneu a fer la mateixa petició al servidor:

```
lxc-attach -n PC01 -- dig @10.0.1.2 -t A www.upc.es
```

- b) Observeu cap diferència en el procés de resolució?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t A www.upc.es
; <>> DiG 9.11.3-1ubuntu1.1-Ubuntu <>> +noadflag +cdflag +qr @10.0.1.2 -t A www.upc.es
; global options: +cmd

;; Sending:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 35958
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.upc.es.           IN      A

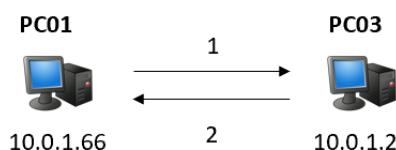
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 35958
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.upc.es.           IN      A

;; ANSWER SECTION:
www.upc.es.          3557    IN      A      147.83.2.135
;; AUTHORITY SECTION:
upc.es.               86357   IN      NS     sun.rediris.es.
upc.es.               86357   IN      NS     ns2.upc.edu.
upc.es.               86357   IN      NS     ns1.upc.edu.
upc.es.               86357   IN      NS     chico.rediris.es.

;; Query time: 0 msec
;; SERVER: 10.0.1.2#53(10.0.1.2)
;; WHEN: Sat Apr 25 11:04:15 UTC 2020
;; MSG SIZE  rcvd: 172
```

La petició que fa el client (PC01) és la mateixa que a l'apartat a) i obté la mateixa resposta del servidor DNS del PC03.

No obstant, si observeu la captura, **Exercici2_apartatB.pcapng**, veureu que el servidor ha donat la resposta al client sense haver de preguntar a cap altra servidor perquè la té a la seva caché.



Reinicieu la captura del *wireshark*. Feu ara la petició DNS següent:

```
lxc-attach -n PC01 -- dig @10.0.1.2 -t MX upc.es
```

- c) Verifiqueu que la resposta obtinguda és l'esperada (fixeu-vos en el camp *flags*, en el contingut de les diferents seccions, etc.) Analitzeu els paquets DNS enviats i rebuts pel servidor com a conseqüència de la petició realitzada. Per cada petició i resposta analitzeu-ne el contingut (fixeu-vos en els flags, en el contingut de les diferents seccions, etc.) Feu un diagrama on aparegui el PC client i tots els servidors DNS que han intervingut en el diàleg i feu un esquema de les peticions i respostes que s'han enviat entre ells. Com afecta en el procés de resolució del contingut del RR tipus MX de *upc.es* que s'hagi fet la petició del RR tipus A de *www.upc.es* anteriorment?

```

root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t MX upc.es
;; <>> DiG 9.11.3-1ubuntu1.1-Ubuntu <>> +noadflag +cdflag +qr @10.0.1.2 -t MX upc.es
;; global options: +cmd

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19232
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:
;upc.es.                      IN      MX

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19232
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:
;upc.es.                      IN      MX

;; ANSWER SECTION:
upc.es.           172800  IN      MX    10 mx2.upc.es.
upc.es.           172800  IN      MX    10 mx1.upc.es.

;; AUTHORITY SECTION:
upc.es.          86326   IN      NS     ns2.upc.edu.
upc.es.          86326   IN      NS     sun.rediris.es.
upc.es.          86326   IN      NS     ns1.upc.edu.
upc.es.          86326   IN      NS     chico.rediris.es.

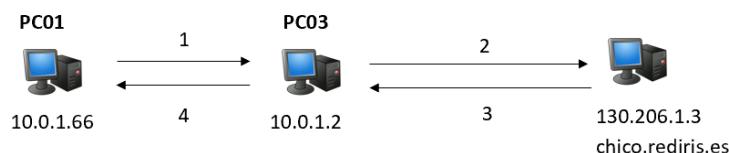
;; Query time: 55 msec
;; SERVER: 10.0.1.2#53(10.0.1.2)
;; WHEN: Sat Apr 25 11:04:46 UTC 2020
;; MSG SIZE  rcvd: 192

```

El client demana el RR tipus MX de `upc.es` al servidor de noms amb adreça IP 10.0.2.1. El client demana recursivitat, flag `rd=1`.

El servidor de noms al que se li fa la petició té la recursivitat habilitada (`ra=1`). El servidor respon que hi ha dos RRs tipus MX a `upc.es`, el primer té el nom `mx2.upc.es` i prioritat 10 i el segon té el nom `mx1.upc.es` i prioritat 10. El servidor de noms no és autoritatius de la resposta (flag `aa=0`) i això vol dir que ha hagut de buscar la resposta en un altre servidor.

A la captura, [Exercici2_apartatC.pcapng](#), podeu veure les peticions i respostes DNS que s'han realitzat des que el client fa la petició fins que en rep la resposta i que estan representants a la figura següent:



A l'apartat a) d'aquest exercici, el servidor 10.0.1.2 ha après la llista de servidors DNS autoritatius del domini `upc.es`. Per tant, quan ara rep la petició del client, tot i que no té la resposta a la pregunta que li fa, no li cal començar a buscar la resposta interrogant un servidor arrel, sinó que pot preguntar directament a un servidor DNS autoritatius de la zona `upc.es` perquè té aquesta informació a la caché i el recurs que vol obtenir està en aquesta zona.

A l'exemple que teniu a la captura, el client 10.0.1.66 fa la petició al servidor 10.0.1.2 i li demana recursivitat (flag `rd=1`) (paquet 1). El servidor 10.0.1.2 no té la resposta, però com que té la recursivitat habilitada, pot fer preguntes a d'altres servidors DNS fins a obtenir la resposta. D'entre la llista de servidors DNS del domini `upc.es` que el servidor 10.0.1.2 té a la caché, escull fer la pregunta al servidor `chico.rediris.es` (paquet 2). El servidor `chico.rediris.es` té la resposta al seu fitxer de zona i envia la resposta al servidor 10.0.1.2 (amb el flag `aa=1`) (paquet 3). Quan el servidor 10.0.1.2 obté la resposta, l'envia al client (amb flag `aa=0`) (paquet 4).

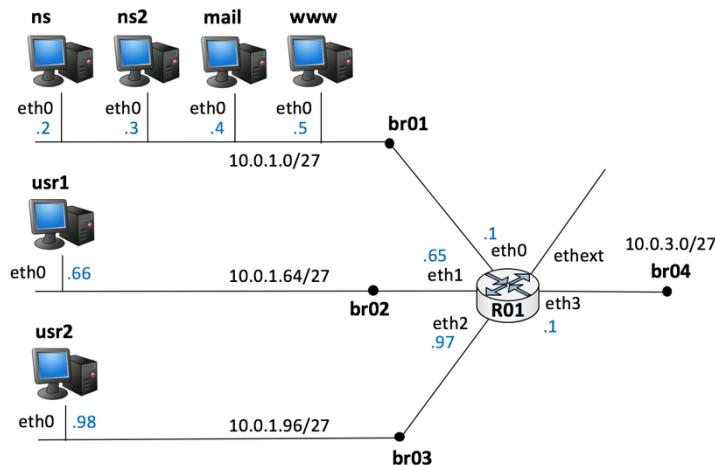
Exercici 3. Configuració d'un servidor mestre d'un domini

A continuació configurareu el PC03 per a que sigui el servidor DNS màster de la zona lab.api i de la zona inversa 1.0.10.in-addr.arpa.

Preparació dels fitxers de zona

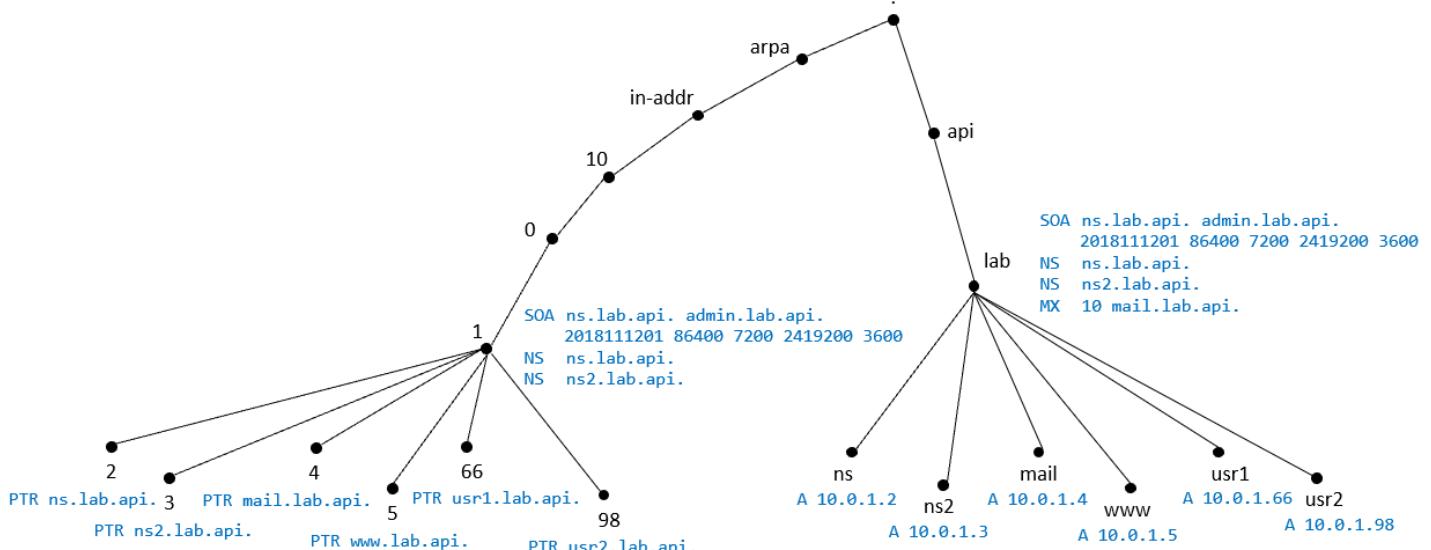
Al fitxer d'una determinada zona hi ha d'haver els RRs (i el seu contingut) que gestiona el servidor DNS autoritatius de la zona. En cada zona hi ha d'haver, com a mínim, un servidor mestre (*master*). Quan s'hagi d'afegir o modificar el contingut d'un RR, caldrà fer-ho al fitxer de zona del mestre. Els servidors esclaus (*slaves*) d'una determinada zona, si n'hi ha, han de descarregar el fitxer de zona del mestre i han d'actualitzar-ne el contingut quan el mestre els ho indica. En principi els fitxers de zona als servidors DNS esclaus no s'han d'editar manualment.

A l'escenari de la pràctica es demana considerar el domini lab.api, que és un subdomini del Top Level Domain (TLD) fictici api. En aquest domini, hi ha d'haver dos servidors DNS: ns.lab.api i ns2.lab.api. El primer (ns.lab.api) serà el PC03 (amb adreça IP 10.0.1.2) i caldrà configurar-lo com a servidor mestre (*master*). El segon (ns2.lab.api) serà el PC04 (amb adreça IP 10.0.1.3) i caldrà configurar-lo com a servidor esclau (*slave*). També caldrà definir un servidor de correu pel domini lab.api (mail.lab.api) que serà el PC05 (amb adreça IP 10.0.1.4) i un servidor web (www.lab.api) que serà el PC06 (amb adreça IP 10.0.1.5). Finalment, caldrà assignar el nom usr1.lab.api al PC01 (amb adreça IP 10.0.1.66) i usr2.lab.api al PC02 (amb adreça IP 10.0.1.98).



- a) Representeu l'espai de noms de la zona lab.api. i de la zona inversa 1.0.10.in-addr.arpa.

L'espai de noms representa (gràficament) la ubicació i el valor dels diferents RR dins el sistema DNS. En aquest cas, ens interessa representar l'espai de noms corresponent a les zones lab.api. i 1.0.10.in-addr.arpa. El resultat final és el que mostra la figura següent:



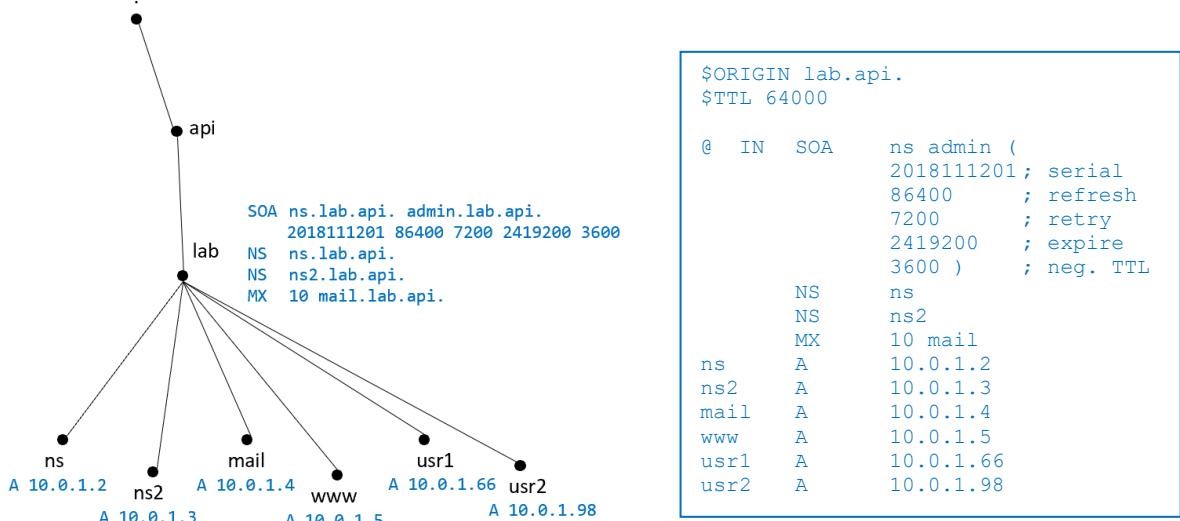
En aquest exemple, a l'arrel de la zona `lab.api.` hi ha el SOA (obligatori per indicar l'inici de la zona), dos RRs tipus NS i un RR tipus MX. La resta de RRs d'aquesta zona són, en aquest cas, RRs tipus A que indiquen les adreces IP assignades a cada nom.

En aquest exemple, a l'arrel de la zona `1.0.10.in-addr.arpa.` hi ha el SOA (obligatori per indicar l'inici de la zona) i els mateixos RRs tipus NS que a la zona `lab.api.` (perquè són els mateixos servidors DNS els que tenen els fitxers de zona de les dues zones). La resta de RRs d'aquesta zona són, en aquest cas, RRs tipus PTR que indiquen els noms assignats a cada adreça IP.

- b) Escriviu els fitxer de zona de la zona `lab.api.` i de la zona inversa `1.0.10.in-addr.arpa.`

Fitxer de la zona `lab.api.`

La figura de l'esquerra representa l'espai de noms on hi ha els RRs de la zona `lab.api.` i a la dreta teniu el contingut del fitxer de la zona `lab.api.`

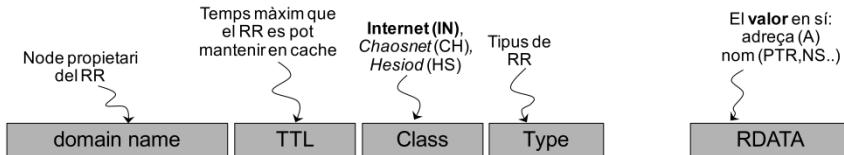


Al principi del fitxer es defineix la variable ORIGIN i se li assigna el nom de l'arrel de la zona (en aquest cas `lab.api.`). Qualsevol nom relatiu que s'escrigui al fitxer (és a dir, qualsevol nom que faci referència a un punt de l'espai de noms que no s'escrigui amb un punt al final) es completarà automàticament amb el valor d'aquesta variable. Per exemple, escriure `ns` (sense punt al final) serà equivalent a escriure `ns.lab.api.` És a dir, els dos fitxers que teniu a continuació són equivalents: el de l'esquerra està escrit amb noms relatius i el de la dreta amb FQDN (Fully Qualified Domain Names). Es poden barrejar noms relatius i noms absoluts al mateix fitxer. Nota: si en la primera columna d'un RR no hi ha res, es considera l'espai de noms del RR de la línia anterior.

<pre>\$ORIGIN lab.api. \$TTL 64000 @ IN SOA ns admin (2018111201 86400 7200 2419200 3600) NS ns NS ns2 MX 10 mail ns A 10.0.1.2 ns2 A 10.0.1.3 mail A 10.0.1.4 www A 10.0.1.5 usr1 A 10.0.1.66 usr2 A 10.0.1.98</pre>	<pre>\$ORIGIN lab.api. \$TTL 64000 lab.api. IN SOA ns.lab.api. admin.lab.api. (2018111201 ; serial 86400 ; refresh 7200 ; retry 2419200 ; expire 3600) ; neg. TTL NS ns.lab.api. NS ns2.lab.api. MX 10 mail.lab.api. ns.lab.api. A 10.0.1.2 ns2.lab.api. A 10.0.1.3 mail.lab.api. A 10.0.1.4 www.lab.api. A 10.0.1.5 usr1.lab.api. A 10.0.1.66 usr2.lab.api. A 10.0.1.98</pre>
--	---

Després de la variable ORIGIN, teniu definida la variable TTL que és el valor de TTL que es fa servir per tots els RRs que no defineixin un valor de TTL explícitament.

Per cada RR que es defineix al fitxer de zona, caldria indicar els camps següents:



El primer RR que teniu al fitxer de l'exemple és el RR tipus SOA (*State-of-Authority*) que ha d'estar obligatòriament al node arrel de la zona (en aquest cas, lab.api.).

```

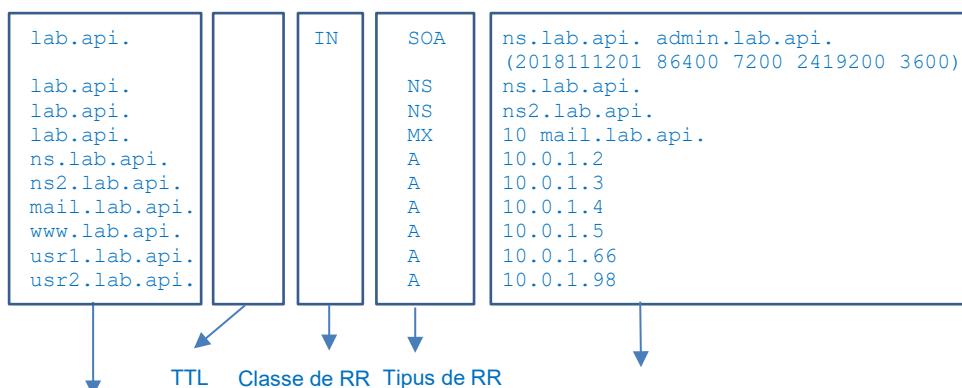
@           IN      SOA
            ns admin (
                2018111201 ; serial
                86400    ; refresh
                7200     ; retry
                2419200  ; expire
                3600 )   ; neg. coach. TTL
  
```

L'@ que apareix a la primera columna del SOA s'expandeix amb el valor de l'ORIGIN del fitxer (en aquest cas, lab.api.). A continuació hi hauria d'haver la columna amb el valor del TTL que, com que no s'indica explícitament, pren el valor indicat a la variable TTL. A la columna de classe de RR, s'indica classe IN (Internet).

El camp RDATA (el contingut) del RR tipus SOA és:

- Nom canònic del servidor màster de la zona
- Adreça de correu electrònic de la persona responsable de la zona
- **Serial.** Número de sèrie del fitxer de zona. Quan un esclau contacta amb el màster, demana el SOA i comprova si el valor del serial que ell té al seu fitxer de zona és inferior al que té el màster. Si és així, significa que les seves dades són obsoletes i ha de demanar la transferència del fitxer de zona. Cada vegada que un administrador modifica el fitxer de zona del màster, ha d'incrementar el valor del serial per tal que els esclaus puguin actualitzar la seva informació correctament. Tot i que l'estàndard no fixa cap format per aquest camp, se sol fer servir el format YYYYMMDDNN, on Y correspon als dígits de l'any actual, M als del mes, D als del dia i N als d'un nombre natural.
- **Refresh.** Indica l'interval de temps que cada esclau ha d'interrogar al màster de la zona per comprovar si s'ha modificat el fitxer. Per defecte s'indica el temps en segons, però també es pot posar la notació en hores (H), dies (D) o setmanes (W).
- **Retry.** Si un esclau no pot contactar amb el màster quan correspon segons el temporitzador de refresh, ho torna a intentar passat un interval de temps indicat per aquest valor. Típicament Retry < Refresh.
- **Expire.** Si un esclau no pot contactar amb el màster després del temps indicat per aquest temporitzador, l'esclau deixa de respondre a peticions sobre el contingut de RRs de la zona.
- **Negative coaching TTL.** És el TTL que s'assigna a respostes negatives (NXDOMAIN).

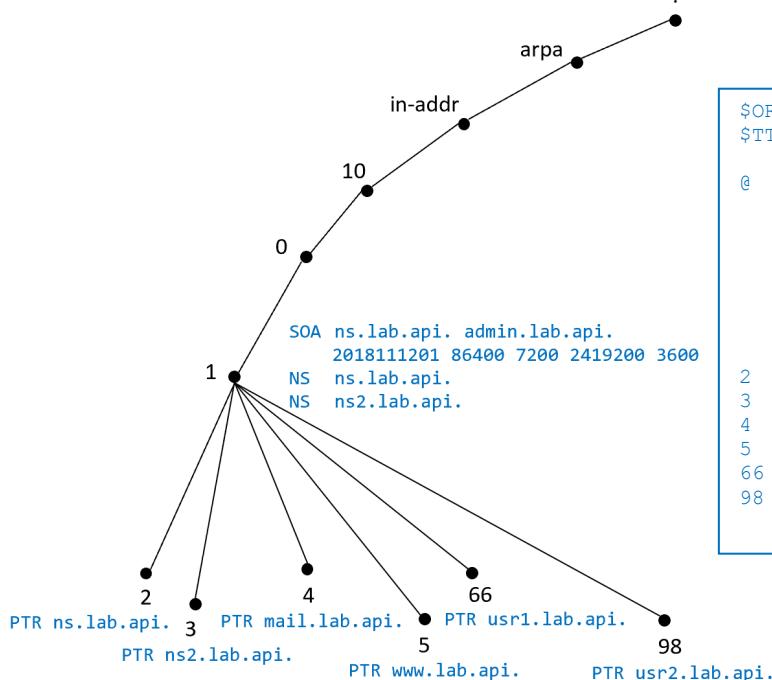
Si no s'especifica un valor de TTL, s'utilitza el de la variable \$TTL. Si s'especifica la classe al primer RR del fitxers (al SOA), la resta de RRs l'hereten. Així doncs, els RRs del fitxer de la zona lab.api. són:



Nom de domini (vèrtex de l'arbre on hi ha el RR, node propietari del RR)

Fitxer de la zona 1.0.10.in-addr.arpa.

La figura de l'esquerra representa l'espai de noms on hi ha els RRs de la zona 1.0.10.in-addr.arpa. i a la dreta teniu el contingut del fitxer de la zona 1.0.10.in-addr.arpa.



```

$ORIGIN 1.0.10.in-addr.arpa.
$TTL 64000

@ IN SOA ns.lab.api. admin.lab.api. (
2018111201; serial
86400 ; refresh
7200 ; retry
2419200 ; expire
3600 ) ; neg. TTL
NS ns.lab.api.
NS ns2.lab.api.
2 PTR ns.lab.api.
3 PTR ns2.lab.api.
4 PTR mail.lab.api.
5 PTR www.lab.api.
66 PTR usr1.lab.api.
98 PTR usr2.lab.api.

```

La variable ORIGIN en aquest cas es defineix amb el valor 1.0.10.in-addr.arpa. i això és el que s'afegirà al final de tots els noms relatius que s'escriuin al fitxer.

El primer RR del fitxer de la zona inversa ha de ser (com en qualsevol fitxer de zona) un RR tipus SOA. En aquest cas, fixeu-vos que al camp RDATA del RR, el nom del servidor i l'adreça de correu s'han d'escriure com a noms absoluts.

Els servidors de noms ns.lab.api. i ns2.lab.api. tenen el fitxer de la zona inversa 1.0.10.in-addr.arpa. Per indicar-ho hi ha dos RRs tipus NS al vèrtex 1.0.10.in-addr.arpa. de l'espai de noms amb el nom d'aquests servidors.

Tots els altres RRs d'aquest fitxer són, en aquest cas, RRs tipus PTR on s'indica quin nom s'ha assignat a cadascuna de les adreces IP del rang 10.0.1.0/24. No surten totes perquè només s'ha assignat un nom a unes quantes adreces IP.

Guardeu els dos fitxers de zona. Podeu triar el nom que vulgueu pels fitxers (els que s'indiquen aquí són els que es fan servir a l'apartat següent).

lab.db

```

$ORIGIN lab.api.
$TTL 64000

@ IN SOA ns admin (
2018111201
86400
7200
2419200
3600 )
NS ns
NS ns2
MX 10 mail
ns A 10.0.1.2
ns2 A 10.0.1.3
mail A 10.0.1.4
www A 10.0.1.5
usr1 A 10.0.1.66
usr2 A 10.0.1.98

```

lab_inv.db

```

$ORIGIN 1.0.10.in-addr.arpa.
$TTL 64000

@ IN SOA ns.lab.api. admin.lab.api. (
2018111201; serial
86400 ; refresh
7200 ; retry
2419200 ; expire
3600 ) ; neg. TTL
NS ns.lab.api.
NS ns2.lab.api.
2 PTR ns.lab.api.
3 PTR ns2.lab.api.
4 PTR mail.lab.api.
5 PTR www.lab.api.
66 PTR usr1.lab.api.
98 PTR usr2.lab.api.

```

Configuració del servidor DNS

Per configurar l'aplicació bind9 per tal que el servidor DNS sigui autoritatius d'una determinada zona, cal seguir els passos següents:

1. Modificar el fitxer `named.conf.local` del directori `/etc/bind` per especificar la zona, el tipus de servidor i el nom del fitxer de zona que conté els RRs.
2. Si el servidor s'ha definit com a màster de la zona, cal crear el fitxer de zona amb els RRs corresponents i guardar-lo amb el nom indicat en el fitxer `named.conf.local` al directori on pertoqui (per defecte, al directori `/var/cache/bind`). Si el servidor s'ha definit com a esclau de la zona, no cal crear ni editar cap fitxer (aquest pas 2 no és necessari).
3. Iniciar (o reiniciar si ja està iniciat) el servei bind9 per aplicar els canvis de configuració.

Pas 1. Modificació del fitxer `named.conf.local`

Editeu el fitxer `named.conf.local` del PC03 amb la comanda:

```
gedit /var/lib/lxc/PC03/rootfs/etc/bind/named.conf.local &
```

Nota: el directori `/var/lib/lxc` és l'arrel de tots els contenidors LXC de l'escenari. Dins de cada contingidor hi ha la carpeta `rootfs`, que és l'arrel del sistema de fitxers propi del contingidor. La comanda anterior obre amb l'aplicació `gedit` el fitxer `named.conf.local` del contingidor PC03. El símbol `&` al final de la comanda, l'executa en *background* per tal de poder seguir executant comandes al mateix terminal sense haver de tancar l'aplicació `gedit`.

Modifiqueu el fitxer per tal d'indicar que el servidor PC03 serà el servidor DNS màster de les zones `lab.api` i `1.0.10.in-addr.arpa`. La sintaxi que heu d'utilitzar és la següent:

```
zone "lab.api" {
    type master;
    file "lab.db";           → Podeu escollir un nom de fitxer diferent, però ha de ser coherent
};                                amb el que s'utilitza al pas 2.  
                                  Si només s'indica el nom del fitxer (com en l'exemple)
                                  caldrà col·locar el fitxer al directori /var/cache/bind del servidor.

zone "1.0.10.in-addr.arpa" {
    type master;
    file "lab_inv.db";       → Podeu escollir un nom de fitxer diferent, però ha de ser coherent
};                                amb el que s'utilitza al pas 2.  
                                  Si només s'indica el nom del fitxer (com en l'exemple)
                                  caldrà col·locar el fitxer al directori /var/cache/bind del servidor.
```

Guardeu els canvis al fitxer.

Verifiqueu que els canvis que heu fet al fitxer `named.conf.local` estan sintàcticament bé amb la comanda:

```
lxc-attach -n PC03 -- named-checkconf
```

Nota: Si el fitxer `named.conf.local` és sintàcticament correcte, no obtindreu cap resposta a l'executar la comanda anterior. Si hi ha algun error sintàctic, se us indicarà en quina línia del fitxer hi ha l'error.

Pas 2. (Només si el servidor DNS s'ha definit com a màster del domini) Creació del fitxer de zona

Com que el PC03 s'ha definit com a servidor màster de les zones lab.api. i 1.0.10.in-addr.arpa cal crear els fitxers d'aquestes zones i guardar-los al directori /var/cache/bind del servidor amb el nom de fitxer que s'hagi indicat al fitxer named.conf.local en el pas anterior.

Creeu el fitxer de la zona lab.api executant la comanda següent:

```
gedit /var/lib/lxc/PC03/rootfs/var/cache/bind/lab.db &
```

Nota: en aquest exemple es considera que al fitxer named.conf.local heu assignat el nom lab.db al fitxer de la zona lab.api.

I ompliu-lo amb el contingut que li correspon a la zona lab.api. (apartat b) d'aquet exercici).

Guardeu els canvis al fitxer.

Verifiqueu que el fitxer és sintàcticament correcte executant les comandes següents:

```
root@api-mv:~# lxc-attach -n PC03
PC03# cd /var/cache/bind
PC03# named-checkzone lab.api. lab.db
```

Nota: Si el fitxer lab.db és sintàcticament correcte, no obtindreu cap resposta a l'executar la comanda anterior. Si hi ha algun error sintàctic, se us indicarà en quina línia del fitxer hi ha l'error.

Creeu el fitxer de la zona 1.0.10.in-addr.arpa. executant la comanda següent:

```
gedit /var/lib/lxc/PC03/rootfs/var/cache/bind/lab_inv.db &
```

Nota: en aquest exemple es considera que al fitxer named.conf.local heu assignat el nom lab_inv.db al fitxer de la zona 1.0.10.in-addr.arpa.

I ompliu-lo amb el contingut que li correspon a la zona 1.0.10.in-addr.arpa. (apartat b) d'aquet exercici).

Guardeu els canvis al fitxer.

Verifiqueu que el fitxer és sintàcticament correcte executant les comandes següents:

```
root@api-mv:~# lxc-attach -n PC03
PC03# cd /var/cache/bind
PC03# named-checkzone 1.0.10.in-addr.arpa. lab_inv.db
```

Nota: Si el fitxer lab_inv.db és sintàcticament correcte, no obtindreu cap resposta a l'executar la comanda anterior. Si hi ha algun error sintàctic, se us indicarà en quina línia del fitxer hi ha l'error.

Pas 3. Reinicieu (o iniciieu) el servidor DNS

Comproveu l'estat del servidor amb la comanda:

```
lxc-attach -n PC03 -- systemctl status bind9
```

Si el servidor està aturat (Active: inactive (dead)), iniciieu-lo amb la comanda:

```
lxc-attach -n PC03 -- systemctl start bind9
```

Si el servidor DNS ja està engegat (Active: active (running)), reinicieu-lo per carregar la nova configuració:

```
lxc-attach -n PC03 -- systemctl restart bind9
```

Verificació del funcionament del servidor

Captureu amb el *wireshark* al bridge br01.

Els paquets que s'analitzen en aquest exercici són els de la captura **Exercici3.pcapng**.

Comproveu el funcionament del servidor DNS utilitzant l'eina **dig**:

- a. Quina petició heu de fer per saber el RR tipus A de `www.lab.api`? Feu la petició, observeu la captura del wireshark i fixeu-vos en el diàleg entre el client i el servidor DNS.

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t A www.lab.api
;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49542
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;www.lab.api.          IN      A
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49542
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; QUESTION SECTION:
;www.lab.api.          IN      A
;; ANSWER SECTION:
www.lab.api.        64000   IN      A      10.0.1.5
;; AUTHORITY SECTION:
lab.api.            64000   IN      NS     ns.lab.api.
lab.api.            64000   IN      NS     ns2.lab.api.
;; ADDITIONAL SECTION:
ns.lab.api.         64000   IN      A      10.0.1.2
ns2.lab.api.        64000   IN      A      10.0.1.3
;; SERVER: 10.0.1.2#53(10.0.1.2)
```

- b. Quina petició heu de fer per saber el nom associat a l'adreça IP 10.0.1.5? Quina resposta obteniu?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t PTR 5.1.0.10.in-addr.arpa.
;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53984
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;5.1.0.10.in-addr.arpa.      IN      PTR
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53984
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; QUESTION SECTION:
;5.1.0.10.in-addr.arpa.      IN      PTR
;; ANSWER SECTION:
5.1.0.10.in-addr.arpa.64000   IN      PTR    www.lab.api.
;; AUTHORITY SECTION:
1.0.10.in-addr.arpa. 64000   IN      NS     ns.lab.api.
1.0.10.in-addr.arpa. 64000   IN      NS     ns2.lab.api.
;; ADDITIONAL SECTION:
ns.lab.api.           64000   IN      A      10.0.1.2
ns2.lab.api.          64000   IN      A      10.0.1.3
;; SERVER: 10.0.1.2#53(10.0.1.2)
```

L'eina *dig* permet fer la mateixa petició indicant directament l'adreça IP per a la qual es vol conèixer el nom que té associat si es fa servir l'opció `-x`. Els paquets 5 i 6 de la captura corresponen a fer la petició següent:

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -x 10.0.1.5
```

- c. Quina petició heu de fer per demanar el servidor de correu del domini lab.api? Quina resposta obteniu?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t MX lab.api
```

```
;; Sending:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46790  
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1  
  
;; QUESTION SECTION:  
;lab.api. IN MX
```

(7)

```
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46790  
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 4  
  
;; QUESTION SECTION:  
;lab.api. IN MX  
  
;; ANSWER SECTION:  
lab.api. 64000 IN MX 10 mail.lab.api.  
  
;; AUTHORITY SECTION:  
lab.api. 64000 IN NS ns2.lab.api.  
lab.api. 64000 IN NS ns.lab.api.  
  
;; ADDITIONAL SECTION:  
mail.lab.api. 64000 IN A 10.0.1.4  
ns.lab.api. 64000 IN A 10.0.1.2  
ns2.lab.api. 64000 IN A 10.0.1.3
```

(8)

- d. Quina petició heu de fer per demanar els RRs del node lab.api? Quina resposta obteniu?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2 -t ANY lab.api
```

```
;; Sending:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29762  
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1  
  
;; QUESTION SECTION:  
;lab.api. IN ANY
```

(9)

```
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29762  
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 4  
  
;; QUESTION SECTION:  
;lab.api. IN ANY  
  
;; ANSWER SECTION:  
lab.api. 64000 IN SOA ns.lab.api. admin.lab.api.  
2018111201 86400 7200 2419200 3600  
lab.api. 64000 IN NS ns2.lab.api.  
lab.api. 64000 IN NS ns.lab.api.  
lab.api. 64000 IN MX 10 mail.lab.api.  
  
;; ADDITIONAL SECTION:  
ns.lab.api. 64000 IN A 10.0.1.2  
ns2.lab.api. 64000 IN A 10.0.1.3  
mail.lab.api. 64000 IN A 10.0.1.4
```

(10)

```
;; SERVER: 10.0.1.2#53(10.0.1.2)
```

Exercici 4. Configuració d'un servidor esclau. Transferències de zona

A continuació, es configurarà el servidor DNS del PC04 com a esclau del servidor DNS del PC03 per a les zones lab.api i 1.0.10.in-addr.arpa.

Per fer-ho, cal seguir els passos següents:

1. Modificar el fitxer named.conf.local del directori /etc/bind tant del màster com de l'esclau.
2. Iniciar (o reiniciar si ja està iniciat) el servei bind9 tant del màster com de l'esclau per aplicar els canvis de configuració.

Pas 1. Modificació del fitxer named.conf.local al màster i a l'esclau

Editeu el fitxer named.conf.local del PC03 amb la comanda:

```
gedit /var/lib/lxc/PC03/rootfs/etc/bind/named.conf.local &
```

Modifiqueu el fitxer per tal d'indicar que el servidor PC03 permetrà la transferència del fitxer de zona al PC04 (amb adreça IP 10.0.1.3) de les zones lab.api i 1.0.10.in-addr.arpa.:

```
zone "lab.api" {
    type master;
    file "lab.db";
    allow-transfer { 10.0.1.3; };
};

zone "1.0.10.in-addr.arpa" {
    type master;
    file "lab_inv.db";
    allow-transfer { 10.0.1.3; };
};
```

Guardeu els canvis al fitxer.

Verifiqueu que els canvis que heu fet al fitxer named.conf.local estan sintàcticament bé amb la comanda:

```
lxc-attach -n PC03 -- named-checkconf
```

Nota: Si el fitxer named.conf.local és sintàcticament correcte, no obtindreu cap resposta a l'executar la comanda anterior. Si hi ha algun error sintàctic, se us indicarà en quina línia del fitxer hi ha l'error.

Editeu el fitxer named.conf.local del PC04 amb la comanda:

```
gedit /var/lib/lxc/PC04/rootfs/etc/bind/named.conf.local &
```

Modifiqueu el fitxer per tal d'indicar que el servidor PC04 serà el servidor DNS esclau de les zones lab.api. i 1.0.10.in-addr.arpa.:

```
zone "lab.api" {
    type slave;
    file "lab.db";           → Podeu escollir un nom de fitxer diferent i no l'heu de crear
    masters {10.0.1.2; };
    masterfile-format text;  → Aquesta comanda us permetrà veure el fitxer de
                                zona en format txt
};

zone "1.0.10.in-addr.arpa" {
    type slave;
    file "lab_inv.db";       → Podeu escollir un nom de fitxer diferent i no l'heu de crear
    masters {10.0.1.2; };
    masterfile-format text;  → Aquesta comanda us permetrà veure el fitxer de
                                zona en format txt
};
```

Guardeu els canvis al fitxer.

Verifiqueu que els canvis que heu fet al fitxer named.conf.local estan sintàcticament bé amb la comanda:

```
lxc-attach -n PC04 -- named-checkconf
```

Nota: Si el fitxer named.conf.local és sintàcticament correcte, no obtindreu cap resposta a l'executar la comanda anterior. Si hi ha algun error sintàctic, se us indicarà en quina línia del fitxer hi ha l'error.

Pas 2. Reinicieu o inicieu el servidor

Per tal d'analitzar el diàleg entre màster i esclau, abans de reiniciar o iniciar els servidors, captureu amb el wireshark al bridge br01.

Reinicieu el servidor DNS al mestre:	<code>lxc-attach -n PC03 -- systemctl restart bind9</code>
Inicieu el servidor DNS a l'esclau:	<code>lxc-attach -n PC04 -- systemctl start bind9</code>

- a) Mirant la captura del wireshark, comenteu el diàleg que s'ha produït entre el mestre i l'esclau.

A la captura [Exercici4_apartatA.pcapng](#) es pot observar el diàleg entre el dos servidors de noms.

En relació a la zona 1.0.10.in-addr.arpa., l'esclau fa una petició al màster demandant el RR tipus SOA de la zona 1.0.10.in-addr.arpa. (paquet 1) i el servidor respon amb el RR tipus SOA de la zona 1.0.10.in-addr.arpa. (paquet 2). Com que l'esclau no té cap fitxer de zona, ha de demanar que el màster li enviï tots els RRs de la zona 1.0.10.in-addr.arpa. Per fer-ho, l'esclau demana el RR tipus AXFR de 1.0.10.in-addr.arpa. (paquet 3). Fixeu-vos que la petició del RR tipus AXFR es fa utilitzant el protocol de transport TCP. El servidor respon amb un paquet DNS on a la secció ANSWER hi ha tots els RRs del fitxer de la zona 1.0.10.in-addr.arpa. (paquet 4)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.1.3	10.0.1.2	DNS	94	Standard query 0xc764 SOA 1.0.10.in-addr.arpa OPT
2	0.000145466	10.0.1.2	10.0.1.3	DNS	214	Standard query response 0xc764 SOA 1.0.10.in-addr.

> Frame 2: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface br01, id 0
 > Ethernet II, Src: 12:75:55:6a:55:15 (12:75:55:6a:55:15), Dst: aa:a2:dc:a1:7a:c2 (aa:a2:dc:a1:7a:c2)
 > Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.3
 > User Datagram Protocol, Src Port: 53, Dst Port: 60546
 > Domain Name System (response)
 Transaction ID: 0xc764
 Flags: 0x8480 Standard query response, No error
 Questions: 1
 Answer RRs: 1
 Authority RRs: 2
 Additional RRs: 3
 Queries
 > 1.0.10.in-addr.arpa: type SOA, class IN
 Answers
 > 1.0.10.in-addr.arpa: type SOA, class IN, mname ns.lab.api
 Authoritative nameservers
 > 1.0.10.in-addr.arpa: type NS, class IN, ns ns.lab.api
 > 1.0.10.in-addr.arpa: type NS, class IN, ns ns2.lab.api
 Additional records
 > ns.lab.api: type A, class IN, addr 10.0.1.2
 > ns2.lab.api: type A, class IN, addr 10.0.1.3
 > <Root>: type OPT
[\[Request In: 1\]](#)

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000905421	10.0.1.3	10.0.1.2	DNS	105	Standard query 0xecb1 AXFR 1.0.10.in-addr.arpa
4	0.001512746	10.0.1.2	10.0.1.3	DNS	342	Standard query response 0xecb1 AXFR 1.0.10.in-addr.

> Frame 4: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface br01, id 0
 > Ethernet II, Src: 12:75:55:6a:55:15 (12:75:55:6a:55:15), Dst: aa:a2:dc:a1:7a:c2 (aa:a2:dc:a1:7a:c2)
 > Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.3
 > Transmission Control Protocol, Src Port: 53, Dst Port: 33031, Seq: 279630870, Ack: 2146994759, Len: 276
 > Domain Name System (response)
 Length: 274
 Transaction ID: 0xecb1
 Flags: 0x8480 Standard query response, No error
 Questions: 1
 Answer RRs: 10
 Authority RRs: 0
 Additional RRs: 0
 Queries
 > 1.0.10.in-addr.arpa: type AXFR, class IN
 Answers
 > 1.0.10.in-addr.arpa: type SOA, class IN, mname ns.lab.api
 > 1.0.10.in-addr.arpa: type NS, class IN, ns ns.lab.api
 > 1.0.10.in-addr.arpa: type NS, class IN, ns ns2.lab.api
 > 2.1.0.10.in-addr.arpa: type PTR, class IN, ns.lab.api
 > 3.1.0.10.in-addr.arpa: type PTR, class IN, ns2.lab.api
 > 4.1.0.10.in-addr.arpa: type PTR, class IN, mail.lab.api
 > 5.1.0.10.in-addr.arpa: type PTR, class IN, www.lab.api
 > 66.1.0.10.in-addr.arpa: type PTR, class IN, usr1.lab.api
 > 98.1.0.10.in-addr.arpa: type PTR, class IN, usr2.lab.api
 > 1.0.10.in-addr.arpa: type SOA, class IN, mname ns.lab.api
[\[Request In: 3\]](#)

En relació a la zona `lab.api.`, l'esclau fa una petició al màster demanant el RR tipus SOA de la zona `lab.api.` (paquet 5) i el servidor respon amb el RR tipus SOA de la zona `lab.api.` (paquet 6). Com que l'esclau no té cap fitxer de zona, ha de demanar que el màster li envii tots els RRs de la zona `lab.api.` Per fer-ho, l'esclau demana el RR tipus AXFR de `lab.api.` (paquet 7). Fixeu-vos que la petició del RR tipus AXFR es fa utilitzant el protocol de transport TCP. El servidor respon amb un paquet DNS on a la secció ANSWER hi ha tots els RRs del fitxer de la zona `lab.api.` (paquet 8)

No.	Time	Source	Destination	Protocol	Length	Info
5	0.500033170	10.0.1.3	10.0.1.2	DNS	82	Standard query 0xcf0d SOA lab.api OPT
6	0.500153445	10.0.1.2	10.0.1.3	DNS	195	Standard query response 0xcf0d SOA lab.api S

> Frame 6: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface br01, id 0
 > Ethernet II, Src: 12:75:55:6a:55:15 (12:75:55:6a:55:15), Dst: aa:a2:dc:a1:7a:c2 (aa:a2:dc:a1:7a:c2)
 > Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.3
 > User Datagram Protocol, Src Port: 53, Dst Port: 53510
 > Domain Name System (response)
 Transaction ID: 0xcf0d
 > Flags: 0x8480 Standard query response, No error
 Questions: 1
 Answer RRs: 1
 Authority RRs: 2
 Additional RRs: 3
 < Queries
 > lab.api: type SOA, class IN
 < Answers
 > lab.api: type SOA, class IN, mname ns.lab.api
 < Authoritative nameservers
 > lab.api: type NS, class IN, ns ns.lab.api
 > lab.api: type NS, class IN, ns ns2.lab.api
 < Additional records
 > ns.lab.api: type A, class IN, addr 10.0.1.2
 > ns2.lab.api: type A, class IN, addr 10.0.1.3
 > <Root>: type OPT
[\[Request In: 5\]](#)

No.	Time	Source	Destination	Protocol	Length	Info
7	0.500659198	10.0.1.3	10.0.1.2	DNS	93	Standard query 0x1fb0 AXFR lab.api
8	0.501063890	10.0.1.2	10.0.1.3	DNS	337	Standard query response 0x1fb0 AXFR lab.api SOA ns

> Frame 8: 337 bytes on wire (2696 bits), 337 bytes captured (2696 bits) on interface br01, id 0
 > Ethernet II, Src: 12:75:55:6a:55:15 (12:75:55:6a:55:15), Dst: aa:a2:dc:a1:7a:c2 (aa:a2:dc:a1:7a:c2)
 > Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.3
 > Transmission Control Protocol, Src Port: 53, Dst Port: 47057, Seq: 1007620549, Ack: 838242365, Len: 271
 > Domain Name System (response)
 Length: 269
 Transaction ID: 0x1fb0
 > Flags: 0x8480 Standard query response, No error
 Questions: 1
 Answer RRs: 11
 Authority RRs: 0
 Additional RRs: 0
 < Queries
 > lab.api: type AXFR, class IN
 < Answers
 > lab.api: type SOA, class IN, mname ns.lab.api
 > lab.api: type NS, class IN, ns ns.lab.api
 > lab.api: type NS, class IN, ns ns2.lab.api
 > lab.api: type MX, class IN, preference 10, mx mail.lab.api
 > mail.lab.api: type A, class IN, addr 10.0.1.4
 > ns.lab.api: type A, class IN, addr 10.0.1.2
 > ns2.lab.api: type A, class IN, addr 10.0.1.3
 > usr1.lab.api: type A, class IN, addr 10.0.1.66
 > usr2.lab.api: type A, class IN, addr 10.0.1.98
 > www.lab.api: type A, class IN, addr 10.0.1.5
 > lab.api: type SOA, class IN, mname ns.lab.api
[\[Request In: 7\]](#)

- b) Mireu i comenteu el contingut dels fitxers que té l'esclau al directori /var/cache/bind

```
root@PC04:/var/cache/bind# more lab_inv.db
$ORIGIN .
$TTL 64000      ; 17 hours 46 minutes 40 seconds
lab.api          IN SOA ns.lab.api. admin.lab.api. (
                  2018111201 ; serial
                  86400       ; refresh (1 day)
                  7200        ; retry (2 hours)
                  2419200    ; expire (4 weeks)
                  3600        ; minimum (1 hour)
)
NS      ns.lab.api.
NS      ns2.lab.api.
MX      10 mail.lab.api.

$ORIGIN lab.api.
mail      A      10.0.1.4
ns        A      10.0.1.2
ns2       A      10.0.1.3
usr1      A      10.0.1.66
usr2      A      10.0.1.98
www       A      10.0.1.5
```

```
root@PC04:/var/cache/bind# more lab_inv.db
$ORIGIN .
$TTL 64000      ; 17 hours 46 minutes 40 seconds
1.0.10.in-addr.arpa IN SOA ns.lab.api. admin.lab.api. (
                  2018111201 ; serial
                  86400       ; refresh (1 day)
                  7200        ; retry (2 hours)
                  2419200    ; expire (4 weeks)
                  3600        ; minimum (1 hour)
)
NS      ns.lab.api.
NS      ns2.lab.api.

$ORIGIN 1.0.10.in-addr.arpa.
2        PTR    ns.lab.api.
3        PTR    ns2.lab.api.
4        PTR    mail.lab.api.
5        PTR    www.lab.api.
66       PTR    usr1.lab.api.
98       PTR    usr2.lab.api.
```

Després de la transferència dels fitxers de zona, el que s'observa comprovant el contingut dels fitxers de zona és que l'esclau té els mateixos RRs que el màster.

Els paquets que es comenten a la resta d'apartats de l'exercici són els de la captura **Exercici4_apartatC.pcapng**

Des del PC01 del vostre escenari demaneu l'adreça IP de `web.lab.api` al mestre i a l'esclau amb l'eina dig.

- c) Obteniu resposta? La resposta és autoritativa? Per què?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag @10.0.1.2 web.lab.api
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 64068
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;web.lab.api.           IN      A
;; AUTHORITY SECTION:
lab.api.          3600    IN      SOA     ns.lab.api. admin.lab.api.
2018111201 86400 7200 2419200 3600
;; SERVER: 10.0.1.2#53(10.0.1.2)

root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag @10.0.1.3 web.lab.api
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 29981
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;;web.lab.api.           IN      A
;; AUTHORITY SECTION:
lab.api.          3600    IN      SOA     ns.lab.api. admin.lab.api.
2018111201 86400 7200 2419200 3600
;; SERVER: 10.0.1.3#53(10.0.1.3)
```

Fent la petició del RR tipus A de `web.lab.api`, el que s'obté, tant si la petició es fa al màster (10.0.1.2) com si es fa a l'esclau (10.0.1.3) és que aquest RR no existeix (status: NXDOMAIN). Els dos servidors responen amb el flag `aa=1` indicant que són autoritatius de la resposta.

La petició al servidor 10.0.1.2 i la resposta són els paquets 1 i 2, respectivament, de la captura **Exercici4_apartatC.pcapng**. La petició al servidor 10.0.1.3 i la resposta són els paquets 3 i 4, respectivament, de la captura.

Reinicieu la captureu del wireshark al bridge br01.

Modifiqueu el fitxer de zona del mestre i afegiu un RR tipus CNAME que sigui `web.lab.api` i que sigui un àlies de `www.lab.api`. Guardeu els canvis (no modifiqueu el número de sèrie del registre SOA).

Per modificar el fitxer de zona del màster, cal:

1. Editar el fitxer:

```
gedit /var/lib/lxc/PC03/rootfs/etc/bind/named.conf.local &
```

2. Afegir el RR que es demana:

```
web.lab.api. CNAME www.lab.api.
```

3. Guardar el fitxer.

Amb el wireshark capturant, reinicieu el servidor DNS del mestre (`systemctl restart bind9`).

- d) Comenteu el diàleg que s'ha produït entre el mestre i l'esclau.

Al reiniciar el servidor, el màster envia un missatge a l'esclau amb `Opcode=4 (zone change notification)` que conté el SOA de la zona `1.0.10.in-addr.arpa.` (paquet 5). Com que el màster envia una petició (flag `qr=0`), l'esclau respon amb un missatge de resposta (flag `qr=1`) que té `Opcode=4` i on no hi ha secció ANSWER (paquet 6).

No.	Time	Source	Destination	Proto	Length	Info
5	145.279016524	10.0.1.2	10.0.1.3	DNS	131	Zone change notification 0x2075 SOA 1.0.10.in-addr.arpa SOA ns.lab.api
6	145.279198658	10.0.1.3	10.0.1.2	DNS	79	Zone change notification response 0x2075 SOA 1.0.10.in-addr.arpa
> Frame 5: 131 bytes on wire (1048 bits), 131 bytes captured (1048 bits) on interface br01, id 0						
> Ethernet II, Src: 12:75:55:6a:55:15 (12:75:55:6a:55:15), Dst: aa:a2:dc:a1:7a:c2 (aa:a2:dc:a1:7a:c2)						
> Internet Protocol Version 4, Src: 10.0.1.2, Dst: 10.0.1.3						
> User Datagram Protocol, Src Port: 52121, Dst Port: 53						
└ Domain Name System (query)						
Transaction ID: 0x2075						
└ Flags: 0x2400 Zone change notification						
0... = Response: Message is a query						
.010 0... = Opcode: Zone change notification (4)						
.... .0. = Truncated: Message is not truncated						
.... .0. = Recursion desired: Don't do query recursively						
....0. = Z: reserved (0)						
....0. = Non-authenticated data: Unacceptable						
Questions: 1						
Answer RRs: 1						
Authority RRs: 0						
Additional RRs: 0						
└ Queries						
> 1.0.10.in-addr.arpa: type SOA, class IN						
└ Answers						
< 1.0.10.in-addr.arpa: type SOA, class IN, mname ns.lab.api						
Name: 1.0.10.in-addr.arpa						
Type: SOA (Start Of a zone of Authority) (6)						
Class: IN (0x0001)						
Time to live: 0 (0 seconds)						
Data length: 40						
Primary name server: ns.lab.api						
Responsible authority's mailbox: admin.lab.api						
Serial Number: 2018111201						
Refresh Interval: 86400 (1 day)						
Retry Interval: 7200 (2 hours)						
Expire limit: 2419200 (28 days)						
Minimum TTL: 3600 (1 hour)						
[Response In: 6]						

No.	Time	Source	Destination	Proto	Length	Info
5	145.279016524	10.0.1.2	10.0.1.3	DNS	131	Zone change notification 0x2075 SOA 1.0.10.in-addr.arpa SOA ns.lab.api
6	145.279198658	10.0.1.3	10.0.1.2	DNS	79	Zone change notification response 0x2075 SOA 1.0.10.in-addr.arpa
> Frame 6: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface br01, id 0						
> Ethernet II, Src: aa:a2:dc:a1:7a:c2 (aa:a2:dc:a1:7a:c2), Dst: 12:75:55:6a:55:15 (12:75:55:6a:55:15)						
> Internet Protocol Version 4, Src: 10.0.1.3, Dst: 10.0.1.2						
> User Datagram Protocol, Src Port: 52121, Dst Port: 53						
└ Domain Name System (response)						
Transaction ID: 0x2075						
└ Flags: 0xa400 Zone change notification response, No error						
1... = Response: Message is a response						
.010 0... = Opcode: Zone change notification (4)						
.... .1.. = Authoritative: Server is an authority for domain						
.... .0. = Truncated: Message is not truncated						
.... .0. = Recursion desired: Don't do query recursively						
.... 0... = Recursion available: Server can't do recursive queries						
.... 0... = Z: reserved (0)						
....0... = Answer authenticated: Answer/authority portion was not authenticated by the server						
....0... = Non-authenticated data: Unacceptable						
.... 0000 = Reply code: No error (0)						
Questions: 1						
Answer RRs: 0						
Authority RRs: 0						
Additional RRs: 0						
└ Queries						
> 1.0.10.in-addr.arpa: type SOA, class IN						
[Request In: 5]						

El màster també envia un missatge a l'esclau amb Opcode=4 (zone change notification) que conté el SOA de la zona lab.api. (paquet 7). L'esclau respon amb un missatge de resposta (flag qr=1) que té Opcode=4 i on no hi ha secció ANSWER (paquet 8).

Com que no s'ha canviat el número de sèrie del fitxer de zona al modificar-lo, l'esclau no detecta que s'hagi produït cap canvi al fitxer del servidor i, per tant, no es sol·licita l'enviament de cap RR:

Des del PC01 del vostre escenari, demaneu el RR tipus A de web.lab.api al mestre i a l'esclau.

- e) Obteniu la mateixa resposta del mestre i de l'esclau? Raoneu la resposta.

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag @10.0.1.2 web.lab.api
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57695
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; QUESTION SECTION:
;web.lab.api.          IN      A
;; ANSWER SECTION:
web.lab.api.        64000   IN      CNAME   www.lab.api.
www.lab.api.        64000   IN      A       10.0.1.5
;; AUTHORITY SECTION:
lab.api.            64000   IN      NS      ns.lab.api.
lab.api.            64000   IN      NS      ns2.lab.api.
;; ADDITIONAL SECTION:
ns.lab.api.         64000   IN      A       10.0.1.2
ns2.lab.api.        64000   IN      A       10.0.1.3
;; SERVER: 10.0.1.2#53(10.0.1.2)

root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag @10.0.1.3 web.lab.api
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 43609
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;web.lab.api.          IN      A
;; AUTHORITY SECTION:
lab.api.            3600    IN      SOA    ns.lab.api. admin.lab.api.
                                         2018111201 86400 7200 2419200 3600
;; SERVER: 10.0.1.3#53(10.0.1.3)
```

Fent la petició del RR tipus A de web.lab.api, al màster (10.0.1.2) s'obté que web.lab.api. és un CNAME de www.lab.api. i que www.lab.api. té un RR tipus A que és 10.0.1.5. Fent la petició a l'esclau (10.0.1.3) el que s'obté és que el RR de web.lab.api. no existeix (status: NXDOMAIN). Els dos servidors responen amb el flag aa=1 indicant que són autoritatius de la resposta.

La petició al servidor 10.0.1.2 i la resposta són els paquets 9 i 10, respectivament, de la captura. La petició al servidor 10.0.1.3 i la resposta són els paquets 11 i 12, respectivament, de la captura.

Reinicieu la captura del wireshark al bridge br01.

Incrementeu el número de sèrie del registre SOA al fitxer del mestre.

Per modificar el fitxer de zona del màster, cal:

1. Editar el fitxer:

```
gedit /var/lib/lxc/PC03/rootfs/etc/bind/named.conf.local &
```

2. Incrementar el número de sèrie. (Fixeu-vos que en l'exemple que s'està utilitzant, el serial del fitxer de zona lab.api. és 2018111201 i n'hi ha prou incrementant una unitat aquest número)

```
@      IN      SOA    ns.lab.api. admin.lab.api.  2018111202 86400 7200 2419200 3600
```

3. Guardar el fitxer.

Amb el wireshark capturant, torneu a reiniciar el mestre (`systemctl restart bind9`).

- f) Comenteu el diàleg que s'ha produït entre el mestre i l'esclau.

Al reiniciar el servidor, el màster envia un missatge a l'esclau amb Opcode=4 (zone change notification) que conté el SOA de la zona `lab.api`. (paquet 13). Com que el màster envia una petició (flag qr=0), l'esclau respon amb un missatge de resposta (flag qr=1) que té Opcode=4 i on no hi ha secció ANSWER (paquet 14).

El número de sèrie del SOA que ha enviat el màster no coincideix amb el que té l'esclau i l'esclau envia un missatge al màster per demanar el RR tipus SOA de la zona `lab.api`. (paquet 15). El servidor respon amb el SOA de la zona `lab.api`. (paquet 16). L'esclau confirma que el número de sèrie del fitxer de la zona `lab.api`. s'ha incrementat respecte el que té ell al fitxer de zona i demana l'enviament de les modificacions del fitxer de zona. Per fer-ho, envia una petició DNS del RR tipus IXFR de la zona `lab.api`. (paquet 17) i rep un missatge de resposta per part del servidor màster amb els RRs de la zona `lab.api`. (paquet 18).

El màster també envia un missatge a l'esclau amb Opcode=4 (zone change notification) que conté el SOA de la zona `1.0.10.in-addr.arpa`. (paquet 19). L'esclau respon amb un missatge de resposta (flag qr=1) que té Opcode=4 i on no hi ha secció ANSWER (paquet 20). Com que no s'ha modificat el fitxer de la zona `1.0.10.in-addr.arpa`, l'esclau no demana res en relació amb aquesta zona.

Des del PC01 del vostre escenari, demaneu el RR tipus A de `web.lab.api` al mestre i a l'esclau.

- g) Obteniu ara la mateixa resposta del mestre i de l'esclau? Raoneu la resposta.

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag @10.0.1.2 web.lab.api
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11638
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; QUESTION SECTION:
;web.lab.api.          IN      A
;; ANSWER SECTION:
web.lab.api.        64000   IN      CNAME   www.lab.api.
www.lab.api.        64000   IN      A       10.0.1.5
;; AUTHORITY SECTION:
lab.api.            64000   IN      NS      ns2.lab.api.
lab.api.            64000   IN      NS      ns.lab.api.
;; ADDITIONAL SECTION:
ns.lab.api.         64000   IN      A       10.0.1.2
ns2.lab.api.        64000   IN      A       10.0.1.3
;; SERVER: 10.0.1.2#53(10.0.1.2)

root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag @10.0.1.3 web.lab.api
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52544
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; QUESTION SECTION:
;web.lab.api.          IN      A
;; ANSWER SECTION:
web.lab.api.        64000   IN      CNAME   www.lab.api.
www.lab.api.        64000   IN      A       10.0.1.5
;; AUTHORITY SECTION:
lab.api.            64000   IN      NS      ns.lab.api.
lab.api.            64000   IN      NS      ns2.lab.api.
;; ADDITIONAL SECTION:
ns.lab.api.         64000   IN      A       10.0.1.2
ns2.lab.api.        64000   IN      A       10.0.1.3
;; SERVER: 10.0.1.3#53(10.0.1.3)
```

Ara sí que fent la petició del RR tipus A de `web.lab.api`, tant al màster (10.0.1.2) com a l'esclau (10.0.1.3) s'obté que `web.lab.api` és un CNAME de `www.lab.api`. i que `www.lab.api` té un RR tipus A que és 10.0.1.5. La petició al servidor 10.0.1.2 i la resposta són els paquets 21 i 22, respectivament, de la captura. La petició al servidor 10.0.1.3 i la resposta són els paquets 23 i 24, respectivament, de la captura.

Exercici 5. Delegació d'un subdomini

En aquest exercici, s'ha de delegar el subdomini `subgrup.lab.api` al servidor DNS que hi ha al PC08. Caldrà delegar també les adreces IP que passen a ser responsabilitat del servidor DNS del subdomini. En aquest cas, el subrang delegat és `10.0.1.128/25`.

La configuració del servidor DNS de la zona `subgrup.lab.api` ja està feta. Acobleu un terminal al PC08 i comproveu el contingut del fitxer `named.conf.local` que hi ha al directori `/etc/bind` i dels fitxers amb extensió `.db` que hi ha al directori `/var/cache/bind`.

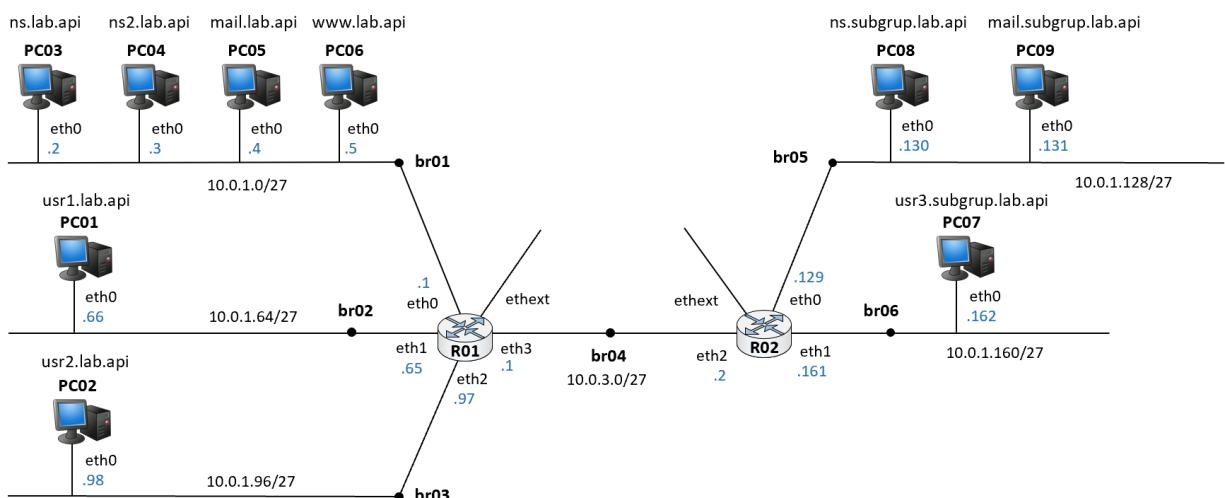
```
PC08# more /etc/bind/named.conf.local
zone "subgrup.lab.api" {
    type master;
    file "dele_lab.db";
};

zone "128/25.1.0.10.in-addr.arpa" {
    type master;
    file "dele_lab_inv.db";
};

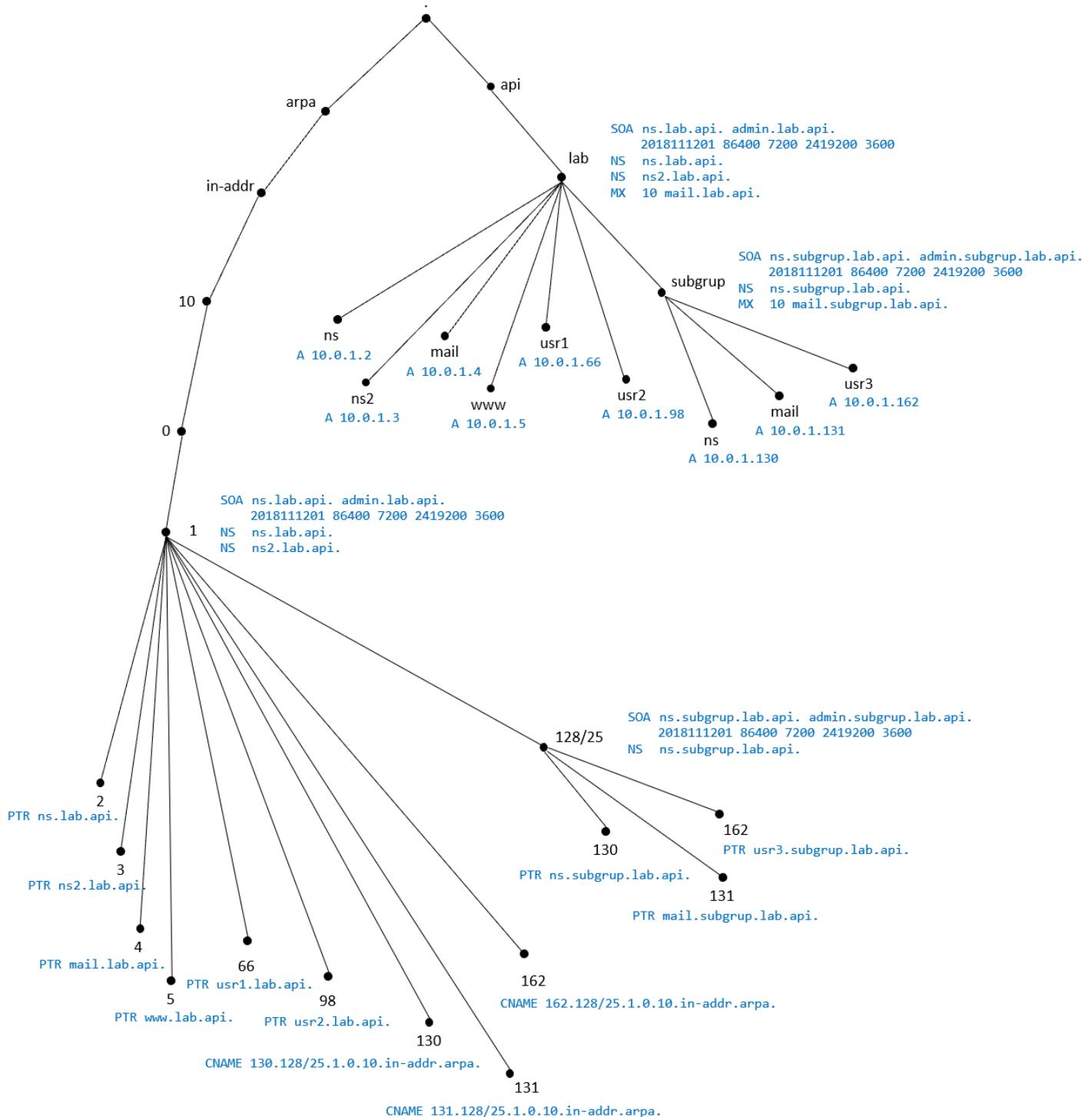
PC08# more /var/cache/bind/dele_lab.db
$ORIGIN subgrup.lab.api.
$TTL 64000
@      IN      SOA     ns      admin  ( 2018111201 1d 2h 4w 1h )
                  NS      ns
                  MX      10 mail
ns      A      10.0.1.130
mail   A      10.0.1.131
usr3   A      10.0.1.162

PC08# more /var/cache/bind/dele_lab_inv.db
$ORIGIN 128/25.1.0.10.in-addr.arpa.
$TTL 64000
@      IN      SOA     ns.subgrup.lab.api.  admin.subgrup.lab.api.
                  ( 2018111201 1d 2h 4w 1h )
                  NS      ns.subgrup.lab.api.
130    PTR    ns.subgrup.lab.api.
131    PTR    mail.subgrup.lab.api.
162    PTR    usr3.subgrup.lab.api.
```

- a) Escriviu el nom que té assignnada cada màquina a la figura de l'escenari.



- b) Afegiu els RRs de les zones del servidor PC08 a l'espai de noms que havíeu representat per a les zones del PC03



Inicieu el servidor DNS del PC08 la comanda:

```
Lxc-attach -n PC08 -- systemctl start bind9
```

Comproveu el correcte funcionament del servidor DNS de R08 fent-li la petició d'algun RR del domini subgrup.lab.api des del PC07.

- c) Quina petició heu fet? Quina resposta heu obtingut?

```
root@api-mv:~# lxc-attach -n PC07 -- dig +noadflag +cdflag +qr @10.0.1.130
-t ANY subgrup.lab.api

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5215
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;subgrup.lab.api.      IN      ANY

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5215
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 3
;; QUESTION SECTION:
;subgrup.lab.api.      IN      ANY
;; ANSWER SECTION:
subgrup.lab.api.    64000   IN      SOA     ns.subgrup.lab.api. admin.subgrup.lab.api.
                           2018111201 86400 7200 2419200 3600
subgrup.lab.api.    64000   IN      NS      ns.subgrup.lab.api.
subgrup.lab.api.    64000   IN      MX      10 mail.subgrup.lab.api.
;; ADDITIONAL SECTION:
ns.subgrup.lab.api. 64000   IN      A       10.0.1.130
mail.subgrup.lab.api. 64000   IN      A       10.0.1.131

;; SERVER: 10.0.1.130#53(10.0.1.130)
```

En aquest exemple s'ha fet la petició de tots els RRs que hi ha la node subgrup.lab.api. i el servidor respon amb els RRs tipus SOA, NS i MX que hi ha al vèrtex subgrup.lab.api.

Tenint en compte que cal delegar la zona subgrup.lab.api. i la zona 128/25.1.0.10.in-addr.arpa., modifiqueu els fitxers de zona del DNS del PC03 per poder fer les delegacions.

lab.db

```
$ORIGIN lab.api.
$TTL 64000

@      IN      SOA     ns admin (
                           2018111201
                           86400
                           7200
                           2419200
                           3600 )
                           NS     ns
                           NS     ns2
                           MX     10 mail
subgrup  NS     ns.subgrup
ns       A      10.0.1.2
ns2      A      10.0.1.3
mail     A      10.0.1.4
www      A      10.0.1.5
usr1     A      10.0.1.66
usr2     A      10.0.1.98
ns.subgrup A      10.0.1.130
```

lab_inv.db

```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 64000

@      IN      SOA     ns.lab.api. admin.lab.api. (
                           2018111201; serial
                           86400      ; refresh
                           7200       ; retry
                           2419200   ; expire
                           3600 )     ; neg. TTL
                           NS     ns.lab.api.
                           NS     ns2.lab.api.
128/25  NS     ns.subgrup.lab.api.
2       PTR    ns.lab.api.
3       PTR    ns2.lab.api.
4       PTR    mail.lab.api.
5       PTR    www.lab.api.
66      PTR    usr1.lab.api.
98      PTR    usr2.lab.api.
130     CNAME  130.128/25
131     CNAME  131.128/25
162     CNAME  162.128/25
```

Al fitxer de la zona `lab.api.`, per poder delegar la zona `subgrup.lab.api.` només cal indicar el nom del servidor DNS on hi haurà el fitxer de la zona `subgrup.lab.api.` i la seva adreça IP:

```
subgrup.lab.api.      NS      ns.subgrup.lab.api.
ns.subgrup.lab.api.  A       10.0.1.130
```

Al fitxer de la zona `1.0.10.in-addr.arpa.`, per poder delegar la zona `128/25.1.0.10.in-addr.arpa.` cal indicar el nom del servidor DNS on hi haurà el fitxer de la zona `128/25.1.0.10.in-addr.arpa.` i, a més a més, cal afegir un CNAME a totes aquelles IPs del subrang `10.0.1.128/25` que ara passaran a dependre del nou servidor DNS:

```
128/25.1.0.10.in-addr.arpa.  NS      ns.subgrup.lab.api.
130.1.0.10.in-addr.arpa.    CNAME  130.128/25
131.1.0.10.in-addr.arpa.    CNAME  131.128/25
162.1.0.10.in-addr.arpa.    CNAME  162.128/25
```

Verifiqueu que els fitxers són correctes i reinicieu el servidor DNS del PC03.

Per verificar la sintaxi dels fitxers de zona, cal executar les comandes següents:

```
root@api-mv:~# lxc-attach -n PC03
PC03# cd /var/cache/bind
PC03# named-checkzone lab.api. lab.db
PC03# named-checkzone 1.0.10.in-addr.arpa. lab_inv.db
```

I per reiniciar el servidor cal executar la comanda:

```
PC03# systemctl restart bind9
```

Captureu amb el wireshark els paquets al br01 i al br05.

Des del PC01 feu una petició DNS al PC03 demanant l'adreça IP de `mail.subgrup.lab.api.`

- d) Quina petició heu fet? Quina resposta heu obtingut?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2
-t A mail.subgrup.lab.api

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52442
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;mail.subgrup.lab.api.      IN      A

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52442
;; flags: qr rd ra cd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;mail.subgrup.lab.api.      IN      A
;; ANSWER SECTION:
mail.subgrup.lab.api. 64000  IN      A      10.0.1.131
;; AUTHORITY SECTION:
subgrup.lab.api.        64000  IN      NS      ns.subgrup.lab.api.

;; SERVER: 10.0.1.2#53(10.0.1.2)
```

- e) Observeu l'intercanvi de missatges de la captura i raoneu el procés que es segueix per obtenir el recurs demanat.

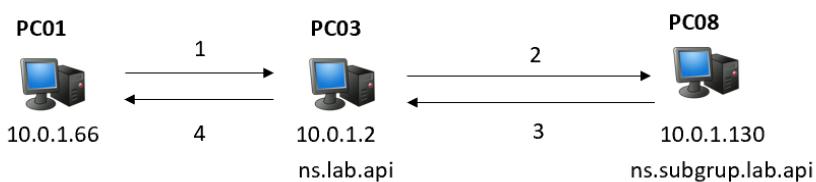
L'intercanvi de paquets des que el client fa la petició al servidor DNS fins que n'obté la resposta és a la captura [Exercici5.pcapng](#).

El client 10.0.1.66 (PC01) demana, amb recursivitat (flag rd=1), el RR tipus A de mail.subgrup.lab.api al servidor 10.0.1.2 (PC03) (paquet 1).

El servidor no té la resposta però té la recursivitat habilitada i, per tant, pot fer la pregunta a altres servidors DNS. El PC03 sap que ha delegat el domini subgrup.lab.api al servidor ns.subgrup.lab.api que té l'adreça IP 10.0.1.130 i, per tant, fa la petició del RR tipus A de mail.subgrup.lab.api al servidor 10.0.1.130 (PC08) (paquet 2).

El servidor 10.0.1.130 té la resposta al seu fitxer de zona, per tant respon que el RR tipus A de mail.subgrup.lab.api és 10.0.1.131 i activa el flag d'autoritat en la resposta (flag aa=1) (paquet 3).

Quan el PC03 rep la resposta, l'envia al client, però amb el flag d'autoritat desactivat (paquet 4).



Des del PC01 feu una petició DNS al PC03 demanant el nom associat a l'adreça IP 10.0.1.131.

- f) Quina petició heu fet? Quina resposta heu obtingut?

```
root@api-mv:~# lxc-attach -n PC01 -- dig +noadflag +cdflag +qr @10.0.1.2
-t PTR 131.1.0.10.in-addr.arpa.
```

```

;; Sending:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37562
;; flags: rd cd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;131.1.0.10.in-addr.arpa.      IN      PTR

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37562
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;131.1.0.10.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
131.1.0.10.in-addr.arpa. 64000      IN      CNAME   131.128/25.1.0.10.in-addr.arpa.
131.128/25.1.0.10.in-addr.arpa. 64000      IN      PTR    mail.subgrup.lab.api.

;; AUTHORITY SECTION:
131.128/25.1.0.10.in-addr.arpa. 64000      IN      NS      ns.subgrup.lab.api.

;; SERVER: 10.0.1.2#53(10.0.1.2)
  
```

- g) Observeu l'intercanvi de missatges de la captura i raoneu el procés que es segueix per obtenir el recurs demanat.

L'intercanvi de paquets des que el client fa la petició al servidor DNS fins que n'obté la resposta és a la captura [Exercici5.pcapng](#).

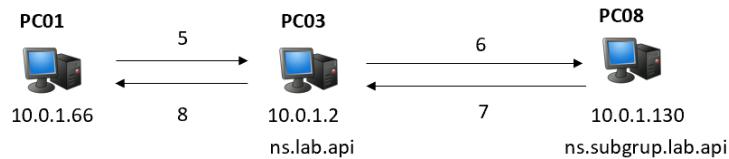
El client 10.0.1.66 (PC01) demana, amb recursivitat habilitada (flag rd=1) el RR tipus PTR de 131.1.0.10.in-addr.arpa al servidor 10.0.1.2 (PC03) (paquet 5).

El servidor té el fitxer de la zona 1.0.10.in-addr.arpa però hi consta que a 131.1.0.10.in-addr.arpa hi ha un CNAME que apunta a 131.128/25.1.0.10.in-addr.arpa. i sap que el servidor DNS que s'encarrega

de la zona 128/25.1.0.10.in-addr.arpa és el servidor subgrup.lab.api. Com que té l'adreça IP del servidor subgrup.lab.api. al fitxer de la zona lab.api., pot fer la petició del RR tipus PTR de 131.1.0.10.in-addr.arpa al servidor 10.0.1.130 (PC08) (paquet 6).

El servidor 10.0.1.130 té la resposta al seu fitxer de zona, per tant respon que el RR tipus PTR de 131.1.0.10.in-addr.arpa és mail.subgrup.lab.api i activa el flag d'autoritat en la resposta (flag aa=1) (paquet 7).

Quan el PC03 rep la resposta, l'envia al client, però amb el flag d'autoritat desactivat (paquet 8).



Quan acabeu la pràctica, atureu l'escenari amb la comanda P08-E01-stop

FIGURES

