Configuració de la xarxa (DNS i DHCP)

Ivan Basart Carrillo i Carles Caño Valls

Serveis de Xarxa

Índex

Introducció

En el mòdul "Serveis de Xarxa" s'estudiarà i es practicarà la instal·lació i configuració de diversos serveis de xarxa com FTP, WEB, SSH, etc. La majoria d'aquests serveis són molt coneguts i destinats a proporcionar servei a l'usuari final (per això són populars). En aquesta unitat tractarem dels serveis DHCP i DNS que, tot i ser omnipresents, no són tan coneguts. Aquests serveis no van destinats a l'usuari final, sinó a la configuració de les xarxes, a fer que les xarxes funcionin (i per això no els coneixem tant).

En la unitat "Configuració de la xarxa (DNS i DHCP)" s'aprèn a mantenir i administrar adequadament els equips en xarxa de manera automatitzada. Es mostra com l'administrador pot establir serveis que permeten configurar automàticament les dades de connexió dels equips i el domini al qual pertanyen i com es configuren equips per actuar de clients i fer ús d'aquests serveis.

En l'apartat "Instal·lació de serveis de configuració dinàmica de sistemes" s'explicarà el funcionament del protocol DHCP. El servei DHCP (*dynamic host configuration protocol*o protocol de configuració dinàmica d'equips) permet la configuració d'adreces IP, màscares, passarel·les per defecte i moltes altres opcions de configuració de manera totalment dinàmica. A cada equip, se li ha de proporcionar un identificador i la informació necessària per poder treballar en xarxa i poder accedir a altres equips i altres xarxes.

Primerament s'analitzarà quina és la informació necessària que ha de tenir un equip per poder treballar en xarxa i disposar d'accés a Internet. Aquesta configuració es pot establir manualment o de manera dinàmica, s'analitzaran els pros i contres de cada cas. El creixement que han sofert les xarxes a escala mundial (tant de nombre de xarxes, d'equips i de complexitat de gestió) va propiciar el sorgiment del prototol DHCP. Se n'estudiarà l'origen, basat en el protocol BOOTP, l'evolució i el funcionament. Així doncs, veurem el clàssic diàleg DHCP d'intercanvi de quatre missatges.

Un cop es conegui la finalitat del protocol DHCP i el seu funcionament, caldrà implementar-ne un servidor. Estudiarem tots els passos necessaris per fer-ho i tots els mecanismes de reconeixement i monitoratge per comprovar que la instal·lació i posada en servei s'ha efectuat correctament.

Finalment caldrà estudiar les opcions de configuració generals del servei i fer un repàs a totes aquelles opcions de xarxa més usuals per als clients.

També es veurà la configuració d'un client DHCP i les opcions que es poden definir directament en el client.

A la unitat "Instal·lació de serveis de resolució de noms" s'aprendrà el protocol DNS (*domain name system* o sistema de noms de domini), que permet la resolució

de noms de domini a adreces IP i a la inversa. La "màgia" amb la qual un usuari indica un nom de domini i s'obté l'adreça corresponent a aquest domini és obra del DNS.

Primerament caldrà familiaritzar-se amb el sistema de noms de domini veient l'evolució, des dels primers fitxers de noms plans, fins a l'actual sistema jeràrquic i distribuït. S'aprendrà a reconèixer i identificar el funcionament dels noms de domini a Internet i com són gestionats a través de servidors encarregats de controlar una zona concreta.

Es presentarà la documentació de l'estàndard del protocol DNS i les diverses extensions que hi ha, que permeten actualitzacions dinàmiques, multitud d'opcions de configuració, configuracions condicionals, expressions i tractament de la seguretat en les comunicacions DNS.

El sistema de noms de domini permet identificar un domini (existent) a qualsevol lloc del món a Internet. Es mostrarà com es realitza aquest mecanisme de resolució, que per art de "màgia" sap identificar quina adreça IP correspon a aquest domini. Aquesta tasca la realitzen els servidors de noms. Caldrà, doncs, instal·lar-lo i posar-lo en funcionament. Es veurà, doncs, tot el procés d'instal·lació necessari per posar en marxa un servidor DNS i els mecanismes de reconeixement que cal utilitzar per comprovarne un funcionament correcte.

Finalment caldrà saber definir noves zones amb informació dels equips propis de la zona. S'explicarà com definir els equips usant els registres de recurs, com compartir aquesta informació entre diversos servidors primaris i secundaris a través de transferències, com delegar zones entre institucions diferents i, fins i tot, com posar en marxa servidors només cau (que no administren res, simplement agilitzen les respostes).

Els dos temes tractats en aquesta unitat, tot i que relacionats, són absolutament independents l'un de l'altre. Recomanem als estudiants fer una primera lectura global del servei DHCP i en una segona lectura anar practicant *in situ* en un servidor els passos que es van descrivint. Aquest procés pràctic es pot ampliar al mateix temps seguint els apunts i les activitats contingudes en el material web. El mateix procediment es pot aplicar per aprendre el funcionament del servei DNS i per practicar la configuració d'un servidor.

Resultats d'aprenentatge

En finalitzar aquesta unitat, l'alumne/a:

1. Instal·la serveis de configuració dinàmica, descrivint les seves característiques i aplicacions.

- Reconeix el funcionament dels mecanismes automatitzats de configuració dels paràmetres de xarxa.
- Identifica els avantatges que proporcionen.
- Il·lustra els procediments i pautes que intervenen en una sol·licitud de configuració dels paràmetres de xarxa.
- Instal·la un servei de configuració dinàmica dels paràmetres de xarxa.
- Presenta el servei per assignar la configuració bàsica als sistemes d'una xarxa local.
- Realitza assignacions dinàmiques i estàtiques.
- Reconeix les diferents topologies de xarxa
- Verifica la correcta assignació dels paràmetres.
- Realitza la documentació adient per donar suport a l'usuari.

2. Instal·la serveis de resolució de noms, descrivint les seves característiques i aplicacions.

- Identifica i descriu escenaris en els que sorgeix la necessitat d'un servei de resolució de noms.
- Classifica els principals mecanismes de resolució de noms.
- Descriu l'estructura, nomenclatura i funcionalitat dels sistemes de noms jeràrquics.
- Instal·la un servei jeràrquic de resolució de noms.
- Prepara el servei per emmagatzemar les respostes procedents de servidors de xarxes públiques i servir-les als equips de la xarxa local.
- Afegeix registres de noms corresponents a una zona nova, amb opcions a servidors de correu i àlies.
- Treballa en grup per realitzar transferències de zona entre dos o més servidors.
- Comprova el funcionament correcte del servidor.
- Realitza la documentació adient per a donar suport a l'usuari.

1. Instal·lació de serveis de configuració dinàmica de sistemes

El servei **DHCP** permet la configuració d'adreces IP, màscares, passarel·les per defecte i moltes altres opcions de configuració de manera totalment dinàmica.

DHCP

DHCP és l'acrònim de *dynamic host configuration protocol*, en català, protocol de configuració dinàmica d'equips.

Què fa el DHCP?

Una forma planera d'entendre el DHCP és imaginar que els equips de client en arrencar fan un crit per la xarxa i pregunten "que hi ha algú?", "qui sóc jo?". El servidor de DHCP els contesta proporcionant-los tota la informació necessària perquè sàpiguen qui són i com han de configurar els seus paràmetres de xarxa.

Identificació dels equips de xarxa

L'administrador de xarxa té la tasca de configurar els equips que la componen. Això significa configurar els servidors, els equips client, concentradors, encaminadors, etc. Cada equip de la xarxa s'ha d'identificar amb l'adreça IP corresponent i la màscara de xarxa, i generalment disposarà d'un camí d'accés a Internet.

Tant els usuaris com els serveis requeriran l'accés a altres equips identificant-los pel nom de domini en lloc de fer-ho per l'adreça IP, que és més difícil de recordar. Fer això equip per equip resulta una feina feixuga i repetitiva si no es disposa de serveis de xarxa que la facilitin.

1.1 El servei DHCP

El servei DHCP proporciona un mecanisme de configuració centralitzat dels equips de la xarxa. En lloc de configurar un per un els equips de xarxa amb adreces i valors estàtics, un servidor DHCP anirà assignant als equips clients els valors que els corresponguin. Aquesta assignació es fa per un període de temps finit, passat el qual caldrà renovar-se.

Els principals avantatges d'utilitzar DHCP són: d'una banda, evitar conflictes d'adreces IP (adreces repetides i adreces errònies), ja que passar equip per equip a canviar la configuració és molt més pesat i propens a l'error que fer-ho editant un sol fitxer de configuració en el servidor DHCP; i, d'altra banda, poder fer l'administració centralitzada representa un estalvi de temps i de feina.

Avantatges DHCP

El servei DHCP té diversos avantatges:

- Evita errors i conflictes IP.
- Centralitza l'administració.
- Estalvia temps.
- Simplifica l'administració.

El servei DHCP simplifica l'administració de la configuració dels equips de xarxa fent-la centralitzada, dinàmica i amb concessions per períodes de temps finits.

La concessió dinàmica d'adreces IP i altres paràmetres de configuració de xarxa es realitza per a un període de temps determinat, que varia en funció de les necessitats del client i del servidor.

Exemples d'ús del servei DHCP

Els següents són alguns exemples d'ús del servei DHCP:

- En una biblioteca que admet connexions Wi-Fi, els clients obtindran concessions per a un temps reduït, per exemple, minuts.
- Un usuari d'Internet que rep al seu equip de casa una adreça IP dinàmica del seu proveïdor d'accés a Internet (ISP) tindrà una concessió que segurament serà per hores.
- En la xarxa corporativa d'una empresa que s'ha configurat dinàmicament usant DHCP, els equips rebran concessions dinàmiques per períodes de temps molt llargs, per exemple, dies.

1.1.1 Configuració d'un equip de xarxa

Qualsevol equip que pertany a una xarxa requereix que es configuri amb uns paràmetres mínims, que són l'adreça IP, la màscara i la porta d'enllaç per defecte. L'adreça IP identifica l'equip de manera única, i la màscara permet determinar la xarxa o subxarxa en què es troba l'equip. Amb aquests dos paràmetres n'hi ha prou per tenir connectivitat en la xarxa. Si es vol disposar d'accés fora de la xarxa pròpia (per exemple, a Internet o a la resta de la xarxa corporativa) cal definir també la porta d'enllaç predeterminada. A part de la configuració bàsica, els equips poden necessitar (i de fet ho necessiten) més paràmetres de configuració com, per exemple: el nom del *host*, el servidor DNS, el fitxer d'iniciació a baixar, etc.

Tot equip de xarxa necessita disposar d'una **adreça IP** que l'identifica de manera única a la xarxa. Cal també una **màscara** per poder separar de l'adreça IP quina és la part d'**adreça de xarxa** i quina la part d'**adreça de host**. Finalment, és imprescindible disposar de l'adreça de la **porta d'enllaç predeterminada** o passarel·la per defecte (o *gateway*), per disposar d'accés a xarxes externes.

Exemple de configuració de xarxa d'un equip de casa:

La majoria d'usuaris disposen a casa d'un equip (o més) connectats a un router que proporciona l'accés a Internet. Aquest equip està configurat com a client DHCP i en iniciar-se rep la configuració de xarxa del router. Podeu comprovar a casa quina configuració teniu. Una configuració d'exemple podria ser:

```
Dirección IP. . . . . . . . . . . : 192.168.1.33
Máscara de subred . . . . . . . . : 255.255.255.0
Puerta de enlace predeterminada   : 192.168.1.1
Servidor DHCP . . . . . . . . . . : 192.168.1.1
Servidores DNS . . . . . . . . . .: 80.58.61.250
                                    80.58.61.254
```

L'inconvenient de la configuració estàtica

La configuració estàtica implica configurar els equips un a un. Fins i tot encara que es tingui accés remot als equips (per Telnet o SSH), com que cal modificar la configuració de xarxa no es pot fer assegut des de l'equip de l'administrador, sinó que cal anar equip per equip a modificar la configuració.

Aquest procés de configuració cal que es faci per a cada equip de la xarxa. Fer-lo manualment implica configurar equip per equip sense cometre errades en teclejar les adreces i les màscares. Qualsevol canvi en l'estructura de la xarxa, com per exemple redefinir les subxarxes o modificar algunes adreces IP, significa tornar a

configurar manualment els equips implicats. És evident que tota aquesta feinada no és agradable per a l'administrador de xarxa (i és molt avorrida!). Tant si la xarxa corporativa consta de pocs equips com de molts, cal una solució que permeti automatitzar la configuració de xarxa de cada equip de manera centralitzada.

Les opcions de configuració de xarxa es poden assignar a cada equip **estàticament** o es poden configurar de manera **dinàmica** utilitzant DHCP.

Com a administradors de xarxa, la gestió centralitzada que ens proporciona DHCP ens permet modificar la xarxa afegint, eliminant i redefinint, equips amb un cost mínim.

1.1.2 Tipus d'assignacions d'adreces IP

Cada equip de xarxa té assignada una adreça IP que l'identifica de manera única dins la xarxa. La composició de l'adreça IP i la màscara determina la xarxa o subxarxa a la qual pertany. A més a més, es configuren altres paràmetres de xarxa com la porta d'enllaç predeterminada, servidors DNS, etc. Això es pot configurar manualment anant equip per equip i introduint aquesta informació.

Reconfiguració d'una xarxa

Imagineu la diversió de l'administrador d'una xarxa corporativa de 1.000 equips amb adreces estàtiques quan cal reconfigurar-la en un cap de setmana!

Quan l'adreça IP i els altres paràmetres necessaris de configuració de la xarxa es configuren equip per equip, manualment, es diu que tenen **IP estàtica**.

Quan la configuració de xarxa d'un equip no es fa manualment i localment en l'equip sinó que es rep per mitjà d'un servidor DHCP, es diu que l'equip utilitza una **IP dinàmica**. Per realitzar configuracions de xarxa dinàmicament caldran un o més servidors de DHCP (a manera de redundància) que proporcionaran la configuració als equips clients (els que cal configurar). Per tant, serà una estructura client-servidor. Les adreces IP dinàmiques que rep el client les podem classificar en dues categories: **assignació dinàmica d'interval** i **assignació fixa**.

El servidor DHCP disposa d'un interval d'adreces que pot assignar als clients que demanen una adreça IP. Quan el servidor assigna una adreça qualsevol de l'interval al client (a l'atzar) es tracta d'una assignació dinàmica d'interval. El client no sap quina adreça IP tindrà i no hi ha manera de predir quina adreça se li concedirà en una futura configuració. A cada nova assignació l'adreça IP pot ser diferent.

MAC

Cada interfície de xarxa s'identifica de manera única físicament per l'adreça MAC (*media acces control*, adreça d'accés al medi).

Una assignació fixa es produeix quan el servidor DHCP sempre assigna la mateixa adreça al client. Per assignar sempre la mateixa adreça IP al client

cal que el servidor pugui identificar inequívocament el client (per l'adreça MAC). El servidor disposa d'una taula amb les correspondències entre les adreces MAC i les adreces IP fixes.

Quan la configuració de xarxa d'un equip es rep per mitjà d'un servidor DHCP es diu que utilitza una adreça IP **dinàmica**. Aquesta adreça pot variar dins d'un **interval** d'adreces disponibles per al servidor DHCP o pot ser **fixa**.

Els avantatges de disposar d'una IP fixa són que la vostra identificació a Internet (la vostra adreça IP) no varia i tothom us pot identificar sempre per la mateixa IP. Podeu proporcionar serveis a altres equips, els clients us identifiquen sempre amb la mateixa adreça sense haver de recordar en cada moment quina adreça IP teniu avui (com passa en el cas d'una IP dinàmica).

DNS dinàmic

Hi ha serveis de DNS dinàmic (DDNS) que permeten assignar un nom de domini a equips amb adreça IP dinàmica.

1.2 El protocol DHCP i els seus components

El protocol DHCP ve descrit, com la majoria de protocols de xarxa, per un document *oficial* anomenat RFC. Al llarg dels anys ha sofert diverses evolucions per anar-se adaptant a les necessitats de cada moment. Tot protocol implica un diàleg entre els equips que hi intervenen, ens caldrà doncs analitzar quin és i com es produeix aquest diàleg. Finalment es descriurà el significat de termes tan usuals en DHCP com *intervals*, *exclusions*, *concessions* i *reserves*.

Per trobar especificacions dels protocols d'Internet, vegeu la secció "Més informació" del web d'aquest mòdul.

Què són els RFC?

Els Request for Comments (RFC) són memoràndums sobre noves investigacions, innovacions i metodologies relacionades amb les tecnologies d'Internet. Quan els publica l'IETF (Internet Engineering Task Force) defineixen a escala mundial els protocols i les seves revisions. És a dir, són les publicacions oficials que descriuen els protocols.

1.2.1 Evolució del protocol DHCP

El servei DHCP és un servei del tipus client-servidor que proporciona la configuració de xarxa als clients que ho sol·liciten. Proporciona els paràmetres bàsics de xarxa com l'adreça IP, la màscara de xarxa, la porta d'enllaç i també d'altres paràmetres necessaris per a una connexió en una xarxa IP. Es tracta d'un protocol de la capa d'aplicació del model TCP/IP.

El protocol DHCP està basat en l'arquitectura de serveis client-servidor i utilitza com a transport el protocol UDP de la pila de protocols TCP/ IP. El servidor DHCP es comunica amb els clients utilitzant paquets UDP, que rep en el seu port 67 i envia al port 68 del client.

La configuració dinàmica d'equips de xarxa es va iniciar amb el protocol BOOTP (BOOT *strap protocol*, protocol d'arrencada). Un protocol més bàsic que principalment permetia definir l'adreça IP, la màscara de xarxa i la passarel·la per defecte per al client. El protocol BOOTP (RFC 951, any 1985) és un protocol pensat per proporcionar automàticament la IP a clients de xarxa en el procés d'arrencada.

L'RFC 951 és el document base que descriu el protocol BOOTP.

Originàriament s'utilitzava per a estacions de treball sense disc que obtenien la configuració de xarxa del protocol BOOTP i també obtenien el nom d'un fitxer d'arrencada que s'havia de baixar per mitjà del TFTP, que usualment era el sistema operatiu.

El BOOTP va donar pas al protocol DHCP, que n'és una evolució amb moltes més prestacions. El DHCP sorgeix l'octubre del 1993 a través de l'RFC 1531. Ràpidament evoluciona per mitjà de diverses RFC, com l'RFC 1541 el mateix any 1993, que serà substituïda per l'RFC 2131 el març del 1997. Aquest document és la base del protocol DHCP actual. A grans trets, el protocol es descriu en l'RFC 2131 per a xarxes Ipv4, el conjunt d'opcions de configuració de DHCP es descriuen en l'RFC 2132, i l'especificació de DHCP per a xarxes Ipv6 és en l'RFC 3315.

RFC de DHCP

Principals RFC dedicades a DHCP:

- RFC 2131. Març 1997. DHCP *dynamic host configuration protocol*.
- RFC 2132. DHCP *options*.
- RFC 3396. *Encoding long options*.
- RFC 4361. *No especific client identifications for dhcpv4*.
- RFC 3315. DHCPv6 *dinamic host configuration protocol Ipv6*.

1.2.2 El model funcional del protocol DHCP

El protocol DHCP descriu el diàleg que es produeix entre client i servidor per a la concessió de configuracions IP. En una xarxa amb configuració d'equips dinàmica, un o més servidors DHCP escoltaran les peticions dels clients en el port 67. Els clients DHCP sol·licitaran al servidor DHCP una configuració IP, i començarà un procés de negociació que ha d'acabar (si tot va bé) amb la concessió d'una adreça IP al client. Els servidors parlen al port 68 dels clients.

Ports DHCP

El protocol DHCP utilitza UDP en la capa de transport. Utilitza dos ports:

- Port 67, on escolta el servidor.
- Port 68, on escolta el client.

La negociació que s'estableix es pot definir a grans trets de la manera següent:

1. El client sol·licita una adreça IP (de fet, una configuració de xarxa).
2. El servidor mira les adreces IP disponibles dins de l'interval d'adreces dinàmiques de què disposa per concedir i n'ofereix una al client.
3. Si el client l'accepta, envia una sol·licitud al servidor per fer-la seva.
4. Si al servidor li sembla bé, accepta la petició del client i li confirma que pot utilitzar aquesta IP, que la hi concedeix per un període de temps limitat.

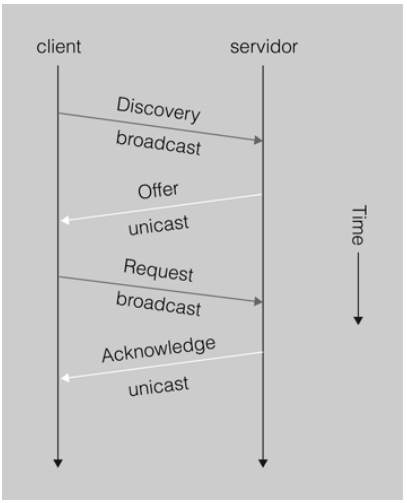
La concessió de l'adreça IP és per un període de temps establert pel servidor. Això significa que, transcorregut aquest període, el client haurà de renegociar la concessió en un procés similar al descrit anteriorment. En la figura 1.1 es pot veure el diàleg de quatre fases client-servidor.

UDP en les transmissions DHCP

L'intercanvi d'informació entre client i servidor no és gaire gran (poc volum de dades) i no requereix un flux permanent (una conversa continuada). És per això que el protocol que s'utilitza en les transmissions DHCP és l'UDP.

El procés real, però, és més detallat. El podem repassar: consta principalment de quatre parts: la petició del client o *discovery*, l'oferta del servidor o *offer*, l'acceptació de l'adreça IP pel client o *request*, i la confirmació del servidor o *acknowledgement*. A part d'aquest tipus de missatges, el protocol DHCP en defineix d'altres com el de petició d'informació o *information* i el d'alliberament de l'adreça IP o *releasing*.

FIGURA 1.1. Model funcional del protocol DHCP



Els següents són els tipus de paquets DHCP:

- DHCP *discover*.
- DHCP *offer*.
- DHCP *request*.
- DHCP *ack* / DHCP *nack*.
- DHCP *decline*.
- DHCP *release*.
- DHCP *information*.

DHCP discover

En un procés de configuració IP d'un client de DHCP el paquet DHCP *discover* és el primer que s'envia. L'envia el client per tal de demanar una configuració IP a algun servidor. Generalment, el client s'acaba d'inicialitzar i vol obtenir una configuració dinàmica de xarxa. El client no sap a quina xarxa pertany (no té adreça IP ni màscara de xarxa) i tampoc sap quins servidors DHCP hi ha en la xarxa (si n'hi ha cap).

Una difusió o broadcast s'adreça a la IP 255.255.255.255 o a l'adreça MAC FF:FF:FF:FF, que és acceptada per tots els equips.

Per tant, el client genera un paquet de difusió (*broadcast*) destinat a tots els equips de la xarxa on sol·licita una configuració IP. En la xarxa pot haver-hi cap, un o més d'un servidor DHCP per atendre aquesta petició. És responsabilitat de l'administrador de xarxes configurar correctament l'estructura i els serveis de xarxa de forma que si defineix clients de DHCP hi hagi servidors DHCP que atenguin les seves peticions.

DHCP offer

En rebre una sol·licitud de configuració d'un client (DHCP *discovery*), un servidor DHCP mira d'atendre-la proporcionant una IP de l'interval d'adreces dinàmiques que gestiona (hi pot haver més d'un servidor DHCP en la mateixa xarxa).

El servidor tracta d'assignar una IP del conjunt o interval (també anomenat *pool*) d'adreces dinàmiques que gestiona. Per fer-ho, ha de mirar quines de les adreces li queden lliures i disponibles per concedir al client. Cada vegada que el servidor concedeix una IP a un client, ho anota en un fitxer de registre de les concessions efectuades. Cada vegada que finalitza una concessió, el servidor pot tornar a utilitzar la IP per a un altre client.

Diversos servidors DHCP

Es pot configurar més d'un servidor DHCP, tant per a còpia de seguretat o *backup* com per incrementar el rendiment en compartir la càrrega de les peticions.

Tota **concessió** (o *lease*) DHCP és per un període determinat de temps i un cop transcorregut cal renovar-la.

El mecanisme que utilitza el servidor per escollir la IP dins del conjunt d'adreces IP disponibles varia en funció del programa de servidor que s'utilitzi. A més a més es poden configurar innumerables opcions del servidor per establir com s'han de fer les concessions. Un cas típic és el de les adreces fixes. A un determinat client sempre se li assigna la mateixa IP. Per això cal disposar de la llista d'adreces MAC dels clients als quals es vol assignar una IP fixa.

El servidor selecciona una IP disponible i la reserva per al client (encara no està assignada). Tot seguit envia un paquet DHCP *offer* (unidestinació o *unicast*) al client amb tota la informació de configuració requerida. L'adreça IP i MAC origen identifiquen el servidor que fa l'oferta. El destinatari s'indica per la seva adreça MAC (que és coneguda). El camp IP del destinatari és l'adreça IP que el servidor ofereix (penseu que el client encara no té IP). Un altre concepte important és per quant de temps es realitza la concessió. El paquet inclou més camps per completar la resta de configuració de xarxa, per exemple, la porta d'enllaç per defecte, els servidors DNS, etc.

Tipus d'adreçament

Hi ha diversos tipus d'adreçament:

- Unidestinació o *unicast*: a un equip
- Multidestinació *multicast*: a un conjunt d'equips
- De difusió o *broadcast*: a tothom.

DHCP request

Quan el client rep una oferta de configuració IP per part d'un servidor, la pot acceptar o rebutjar. Si el client no accepta l'oferta, simplement realitzarà un DHCP *discovery* de nou. Això és suficient perquè el servidor s'adoni que l'oferta ha estat rebutjada.

Si el client accepta l'oferta, ho ha de comunicar al servidor. El mecanisme per fer-ho és mitjançant un paquet DHCP *request* enviat un altre cop per difusió. A hores d'ara, el client encara no disposa de l'adreça IP per utilitzar-la. El servidor l'ha reservat, però encara no ha donat el sí definitiu perquè sigui concedida al client.

El motiu pel qual el client demana quedar-se la concessió (DHCP *request*) que ha rebut utilitzant difusió és fer públic a tothom de la xarxa que ha acceptat una oferta d'un servidor DHCP concret. Recordeu que la petició del client es fa per

difusió i, per tant, pot rebre ofertes de diferents servidors DHCP. Quan accepta una de les ofertes, no ha de dir res als altres servidors que ha refusat. Simplement fent pública quina oferta accepta, la resta de servidors DHCP entenen que la seva oferta s'ha rebutjat.

DHCP ack/DHCP nack

ACK i NACK

ACK i NACK són dos acrònims usuals en el món de la informàtica que signifiquen confirmació (acceptació) i no-conformitat (refús) respectivament.

L'últim pas en una negociació DHCP bàsica el realitza el servidor quan finalment autoritza la concessió enviant el paquet DHCPACK (DHCP *akcnowledgement*). A partir d'aquest moment, el client sí pot fer ús de l'adreça IP i de la configuració de xarxa rebuda. DHCPACK inclou tota la informació referent a la durada de la concessió i les dades necessàries per gestionar quan expira.

El servidor anotarà en el registre de concessions la que acaba de realitzar i detallarà tots els aspectes d'aquesta, en especial el temps de concessió. El paquet d'acceptació de la concessió DHCPACK és un paquet unidestinació adreçat a la MAC del client. Recordeu que el client encara no disposa d'una adreça IP vàlida, en disposarà en rebre el DHCPACK.

Exemple de mala configuració d'un equip

Un exemple de mala configuració és la d'un equip que s'ha engegat amb una adreça IP estàtica errònia que se solapa amb les adreces IP que reparteix el servidor. El mecanisme usat per comprovar si l'adreça IP ja està essent utilitzada és fer un ping. Si ningú no respon és que està lliure (segurament).

Quan un servidor DHCP detecta que la IP que havia reservat per a un client i que li anava a concedir ja està en ús, el servidor envia al client un paquet DHCPNACK i indica la no-autorització de la concessió. El client que rep un DHCPNACK ha de tornar a iniciar tot el procés de negociació començant un altre cop pel DHCP *discovery*.

Com és possible que algun equip de la xarxa utilitzi una adreça IP que forma part del conjunt d'adreces IP dinàmiques que reparteix el servidor DHCP? La resposta és senzilla: perquè hi ha algun equip mal configurat.

DHCP decline

Per la seva part, el client també pot examinar l'adreça IP oferta pel servidor per comprovar si està en ús o no. Pot fer altres proves per veure si li sembla correcta o no l'oferta rebuda del servidor. Per exemple, en el cas de renovació d'una IP el client pot rebre una IP diferent a la que utilitza i no li interessa. En aquests casos, el client pot enviar un paquet DHCP *decline* al servidor per indicar que la seva oferta ha estat rebutjada.

DHCP release

Quan un client ja no necessita més l'ús de la configuració IP que ha rebut, la pot alliberar enviant al servidor un paquet DHCP *release*. En fer-ho, el servidor afegeix l'adreça IP al conjunt d'adreces dinàmiques que té disponibles. També fa l'anotació pertinent en el registre de concessions (*leases*) per indicar que ha finalitzat l'ús de l'adreça. De totes maneres, molt sovint el client no pot arribar a emetre aquest paquet perquè és apagat per l'usuari sense deixar temps al sistema per alliberar la IP.

DHCP information

En tot moment el client pot sol·licitar més informació sobre la configuració de xarxa al servidor utilitzant un paquet DHCP *information*. En el paquet DHCP *offer* que el servidor envia al client, consten les informacions generals de configuració de xarxa que es trameten en l'oferta: adreça IP, màscara de xarxa, porta d'enllaç predeterminada, servidor DNS, fitxer a baixar i molts altres paràmetres que poden estar configurats per enviar-se en l'oferta. El client pot tornar a demanar al servidor la informació d'aquests paràmetres o pot sol·licitar informació per a la configuració d'altres paràmetres (WINS, NetBIOS, *hostname*, etc.). El client només pot realitzar una petició d'informació DHCP *information* al servidor un cop ja està configurat.

Petició de renovació/concessió d'una IP concreta

El procés de quatre fases usuals de DHCP consistent en *discovery/offer/request/ack* es produeix quan el client sol·licita una IP de nou. Sabem que les concessions són per a un interval de temps passat, la renovació de la qual cal que el client la demani. Existeix, doncs, un procés de renovació simplificat. El client demana continuar usant la mateixa IP amb un paquet DHCP *request*, i el servidor li concedeix o no amb els paquets DHCP ACK/NACK.

Un altre cas és un client que demana usar (renovar) una adreça IP que el servidor no li pot concedir (està en ús, no és de l'interval que gestiona, etc.). En aquesta situació, el servidor envia un DHCP NACK.

Macchange

L'ordre GNU permet (*mascarade*) emmascarar l'adreça MAC pròpia. Compte! No feu dolenteries.

1.2.3 Atacs al funcionament de DHCP i conflictes d'adreces IP

vei. Consisteix a inundar de peticions un servidor per tal de saturar-lo i bloquejar-ne el funcionament. Un client pot realitzar innumerables peticions DHCP *discovery* fingint que són clients diferents (emmascarant la seva MAC) amb la intenció d'esgotar les adreces IP disponibles del servidor o simplement amb la intenció de sobrecarregar-lo amb tantes peticions que no doni a l'abast a atendre-les o que ho faci lentament.

Un altre tipus d'atac consisteix a falsejar la informació que s'envia al client. Recordem que el client fa una sol·licitud d'IP en forma de difusió (*broadcast*) i la seva petició pot ser atesa per un o més servidors DHCP. Un dels servidors DHCP pot ser un atacant que intentarà proporcionar informació de configuració falsa al client. Per exemple indicant un servidor DNS també maliciós. Aquest pot falsejar les identitats de les màquines de la xarxa i, quan el client s'adreça a la seva entitat bancària, el servidor DNS en realitat li ha proporcionat una IP d'una màquina que falseja la de l'entitat bancària. Perillós, oi?

Tipus d'atacs DNS

Clients no autoritzats: accés a servidors DNS per part de clients no autoritzats. Servidors no autoritzats: servidors DNS impostors que suplanten els vertaders servidors.

Per posar remei a la inseguretat en la comunicació client-servidor DHCP, el protocol permet utilitzar mecanismes d'autenticació i xifratge. Aquests mecanismes queden fora de l'abast d'aquesta explicació.

Conflictes amb les adreces IP

Un dels principals motius per utilitzar DHCP és simplificar el procés de configuració de xarxa i minimitzar els conflictes per encavalcament d'adreces IP. Per desgràcia, això no garanteix que no es puguin produir conflictes. Per exemple, ens podem trobar en situacions en què dues màquines diferents tinguin la mateixa IP per una simple mala configuració del servidor DHCP. Un altre cas típic és el d'un client que s'ha configurat ell mateix una IP estàtica quan en la xarxa ja hi havia un equip que utilitzava la mateixa adreça IP assignada pel servidor DHCP.

Un problema habitual per als administradors poc experimentats és definir una configuració de xarxa local al client (*hostname*, servidor DNS, porta d'enllaç a utilitzar, etc.), però demanar l'adreça IP dinàmicament. La configuració dinàmica no és solament la IP i la màscara, sinó que el servidor DHCP pot proporcionar altres paràmetres de xarxa que sobreescriuran els que el client tenia definits localment (aquest és l'objectiu de DHCP!).

La configuració rebuda per DHCP sobreescriu la configuració local del client.

1.2.4 Intervals, exclusions, concessions i reserves

Els clients DHCP obtenen del servidor una configuració de xarxa. Descrivim ara alguns dels termes que tenen lloc durant aquest procés, i que formen part de la configuració DHCP.

- **Interval**: anomenen *interval d'adreces IP* el conjunt d'adreces dinàmiques que el servidor té disponibles per assignar als clients. Les adreces IP disponibles s'agrupen per oferir-se a les diverses subxarxes que atén el servidor. Una mateixa subxarxa pot disposar de diversos intervals. Segurament s'entendrà més fàcilment amb un exemple:

```
subnet 140.220.191.0 netmask 255.255.255.0 {
              range 140.220.191.150 140.220.191.249;
            }

  subnet 239.252.197.0 netmask 255.255.255.0 {
              range 239.252.197.10 239.252.197.107;
              range 239.252.197.113 239.252.197.250;
            }
```

En l'exemple anterior s'observa que la primera subxarxa disposa d'un interval de 100 adreces dinàmiques (de la 140.220.191.150 al 250). La segona subxarxa permet assignar dinàmicament dos intervals d'adreces no correlatius.

- **Exclusions**: entenem per *exclusions* aquelles adreces IP que no s'ofereixen dinàmicament per part del servidor. És a dir, que no formen part de cap interval.

- **Concessions**: l'assignació d'una adreça IP i la resta de paràmetres de xarxa a un client per part del servidor, és una concessió (o *leasse*). Els clients reben les concessions per períodes de temps finits, que en finalitzar, cal renegociar. Tant el client com el servidor s'anoten les concessions, el client la que rep i el servidor les que concedeix. Quan finalitza una concessió el servidor pot decidir revocar-la o ampliar-ne la concessió.

El client tothora pot decidir renunciar a la concessió. Si el client vol allargar la concessió inicia un diàleg DHCP abreujat amb el servidor que pot acabar amb una renovació o amb la pèrdua de la concessió (sempre pot tornar a començar el procés). Tant el servidor com el client normalment miren les concessions que s'han efectuat entre ells amb anterioritat per tal de, si és possible, repetir la mateixa assignació.

- **Reserves**: anomenem *reserves* aquelles adreces IP que s'assignen via DHCP però de manera fixa. És a dir, són adreces que s'assignen dinàmicament però sempre i únicament a un *host* determinat. Fixeu-vos que tot i ser una adreça dinàmica només s'utilitza si el *host* associat en fa ús. Si el *host* està apagat l'adreça no es pot usar per a altres *hosts*, està reservada. Un exemple podria ser:

```
subnet 140.220.191.0 netmask 255.255.255.0 {
       host iocserver {
           hardware ethernet 08:00:2b:4c:59:23;
           fixed-address 140.220.191.1;
       }
      range 140.220.191.150 140.220.191.249;
       }
```

En aquest exemple es pot veure que l'adreça *140.220.191.1* és una adreça reservada exclusivament per al *host* iocserver, que s'identifica mitjançant la seva adreça MAC.

1.2.5 DHCP un servei client/servidor

Com acostuma a passar amb els serveis client/servidor, un equip pot realitzar les dues funcions al mateix temps.

El servei DHCP és un més dels serveis de xarxa que tenen l'estructura client /servidor. Els servidors DHCP són els equips que tenen en execució el programa servidor. És el programa encarregat d'atendre les peticions dels clients i oferir-los la configuració de xarxa, tot portant el registre de les IP que concedeix i el registre de totes les accions que realitza. Els clients DHCP són aquells equips que realitzen peticions per obtenir una configuració de xarxa a un servidor DHCP.

El client DHCP

ISP

ISP: *Internet service provider* o proveïdor de servei/accés a Internet. Ho són per exemple les empreses Ono, Vodafone, Jazztel, etc.

Un equip client DHCP és un equip que sol·licita la IP i altres paràmetres de configuració de xarxa a un servidor DHCP en lloc de tenir-los definits localment en l'equip.

Si connecteu el vostre equip informàtic a la xarxa Internet per mitjà d'un ISP (*Internet service provider* o proveïdor de servei/accés a Internet), segurament rebreu una IP dinàmica del vostre proveïdor. Quan es realitzava una trucada telefònica amb mòdem i usant el protocol PPP (*point to point protocol*, protocol punt a punt), el proveïdor proporcionava una adreça IP dinàmica. Si utilitzeu ADSL i un encaminador o *router*, segurament l'encaminador us proporciona una adreça IP dinàmica privada a l'ordinador de casa. Al mateix temps, l'encaminador obté una IP dinàmica pública del proveïdor. Aquestes adreces IP dinàmiques són fixes (sempre les mateixes) o dinàmiques d'interval (pot ser qualsevol IP del conjunt d'adreces IP que té disponibles per concedir el servidor DHCP).

IP pública / IP privada

La diferència entre una IP pública i una IP privada és que la pública és visible per a tots els equips d'Internet, mentre que la privada és visible només dins de la mateixa xarxa local.

El client DHCP ha de tenir en funcionament un dimoni (*daemon*) encarregat de la gestió de les tasques DHCP per part del client. El programa client realitza la part de negociació encarregada al client (DHCP *discovery*, *request*) i també porta un registre de les concessions (*leases*) rebudes. Aquest registre és el que utilitza el client per tornar a demanar la mateixa IP que tenia anteriorment. Un cop rebuda la concessió el programa client queda "adormit", pendent de tornar-se a executar automàticament quan calgui renegociar la concessió. Sense intervenció de l'usuari, el programa client s'activa i segueix el procediment necessari per renegociar l'adreça IP cada cop que el temps de la concessió s'exhaureix.

Configuració client

Usualment les configuracions clients es poden fer de tres maneres diferents:

- fitxer de text: editar directament els fitxers de configuració.
- menús en mode text: usant algun programa de menús amb interfície de text.
- aplicació gràfica: usant una aplicació de finestres en l'entorn gràfic.

Els programes client varien d'un sistema operatiu a un altre i la manera d'executar-los també. Generalment es disposa d'un client executable en mode text o ordres i d'una interfície gràfica (GUI, *graphics user interface* o interfície gràfica d'usuari) per a la configuració. No cal dir que els sistemes Windows tendeixen a la configuració gràfica usant finestres i a la configuració i execució interna d'amagat de l'usuari. Normalment, en els sistemes GNU/Linux, la configuració es fa usant fitxers de text o com a opcions en l'ordre d'execució. La interfície gràfica acostuma a ser un frontal (*front-end*) per cridar l'ordre. Segons sigui el sistema operatiu es pot consultar el fitxer de registre de les concessions rebudes pel client, el fitxer de *leasses*, més o menys detalladament.

Generalment el programa client es pot configurar per definir-ne el comportament en la comunicació amb el servidor: informació a demanar, informació a proporcionar al servidor, opcions per defecte, etc.

1.2.6 El servidor DHCP

L'administrador de xarxa és l'encarregat de pensar la ubicació del servidor o servidors DHCP en l'estructura corporativa. Com més complicada sigui la topologia de la xarxa, més difícil en serà la gestió. Una xarxa corporativa bàsica

pot disposar d'un únic servidor DHCP que ofereix els seus serveis a tots els equips de la xarxa. Els clients poden estar en una mateixa subxarxa o en diverses subxarxes, però totes amb connectivitat amb el servidor DHCP. Aquest també pot ser l'esquema d'una xarxa privada a casa, on un encaminador (el de l'ISP, per exemple) proporciona el servei DHCP a tots els ordinadors de la casa.

Si la xarxa corporativa creix i passa a tenir subxarxes segmentades amb tallafocs les unes de les altres, la configuració del servidor DHCP es complica. Si es vol continuar amb un únic servidor per a tota la xarxa, caldrà que els tallafocs (*firewalls*) deixin passar els paquets DHCP entre les subxarxes i el servidor. Una altra opció és posar un servidor DHCP per a cada subxarxa o grups de subxarxes. Fent-ho així, l'administració de cada servidor és més senzilla però hi ha més servidors a administrar. Una xarxa amb una casuística completa és la que té diversos servidors DHCP per a diverses parts de la xarxa i tallafocs entre clients i servidors que han de permetre el pas de paquets DHCP.

Si el servidor DHCP és l'encarregat de donar adreces IP als clients, qui li proporciona una adreça IP a ell? O bé un altre servidor DHCP (i podríem tornar a fer la mateixa pregunta indefinidament) o bé l'administrador. Usualment, en una xarxa corporativa el servidor DHCP utilitza una IP estàtica definida per l'administrador. Això li permet estar sempre disponible per als clients amb la mateixa IP i no el fa dependre d'un altre servidor extern.

Hi ha diversos programes servidors DHCP: en mode text, en mode gràfic, en mode "màgic" (no es veu què fan) i en mode Unix (tot basat en fitxers de text). Cada administrador treballa amb les seves eines preferides. Les tasques bàsiques per aprendre a utilitzar un servidor DHCP són observar/fer una llista de la configuració actual, activar/aturar el servei, modificar la configuració, monitorar els *logs* (registre de successos del servei) i, evidentment, saber instal·lar i desinstal·lar l'aplicació servidor.

Els fitxers de logs (successos) recullen els esdeveniments que es volen monitorar.

Com la majoria de serveis de xarxa, el servei DHCP és un servei que s'executa en segon pla en forma de dimoni. El servidor DHCP sempre està engegat escoltant en el port 67 les peticions que rep dels clients. Quan rep una petició entrant, el programa executable del servidor DHCP la processa i posa en marxa tot el mecanisme DHCP pertinent per tornar a escoltar noves peticions. De fet, el servidor sempre escolta peticions i les processa simultàniament (segons la configuració).

Els fitxers del registre del servei, on s'anoten les concessions, permeten mantenir la informació encara que el servei s'aturi o que el servidor s'apagui. En tornar a engegar es llegiran de nou els fitxers de registres per tal de saber quines són les concessions que s'havien realitzat.

Els fitxers de concessions permeten mantenir la coherència de l'assignació d'adreces IP entre aturades del servei.

1.3 Instal·lació d'un servei DHCP

El servei de xarxa DHCP està estructurat en forma de servei client / servidor; per tant, caldrà disposar del programari apropiat per fer cada un d'aquests rols. El programari que fa la funció de client usualment ja està integrat en el sistema operatiu (la part que gestiona la xarxa). És a dir, per disposar de la part client del servei DHCP normalment no cal instal·lar res, ja forma part del servei de xarxa.

Així doncs, quan parlem d'instal·lar un servei DHCP fem referència al procés d'instal·lació i configuració del programari del servidor DHCP. Evidentment també caldrà configurar els clients adequadament per fer ús d'aquest servei.

La instal·lació del programari que proporciona el servei DHCP es fa de manera molt similar (per no dir idèntica) al programari d'altres serveis de xarxa com els serveis DNS, HTTP, FTP, etc. Es tracta d'instal·lar el programari de l'aplicació servidor i fer-ne la configuració apropiada. Senzill oi?

Per fer això cal plantejar-se els passos següents:

1. Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
2. Obtenir l'aplicació que proporciona el servei DHCP.
3. Observar l'estat de la xarxa actual. Està el servei ja en funcionament? Existeix ja una configuració DHCP activa?
4. Instal·lar l'aplicació servidor.
5. Comprovar que la instal·lació s'ha efectuat correctament.
6. Configurar el servei en el servidor i activar els clients perquè l'utilitzin.
7. Comprovar que el servei funciona correctament.

1.3.1 Aplicacions servidor DHCP

Sempre que l'administrador vol posar en funcionament un nou servei de xarxa cal que primerament analitzi quines aplicacions hi ha al mercat que ofereixen aquest servei. És feina seva estudiar les característiques de les diverses aplicacions, com per exemple: avaluar-ne l'eficiència, el cost, el que en diuen els altres... La manera més fàcil de fer això és navegar per Internet, consultar les revistes especialitzades o demanar consell a un dels gurus informàtics coneguts.

Usualment l'administrador s'informa a través del seu cercador preferit, per exemple Google, i de webs com la Viquipèdia. Podeu cercar DHCP o DHCP server al Google i a la wiki (en anglès).

Usualment, però, l'administrador acaba utilitzant l'aplicació servidor DHCP que li proporciona el mateix sistema operatiu. Si utilitzeu el sistema operatiu Windows l'empresa Microsoft disposa d'una aplicació pròpia, però també en podeu trobar

d'altres a Internet. Igualment si utilitzeu GNU/Linux segurament la mateixa distribució ja proporciona un servidor DHCP. De totes maneres en podeu obtenir d'altres també a Internet.

1.3.2 Instal·lar l'aplicació servidor

Tot seguit es descriurà el procés per instal·lar el servei DHCP en un entorn GNU/Linux. Un cop feta la instal·lació cal observar què s'ha instal·lat, quins programes executables, on són els fitxers de configuració, els de monitoratge, etc.

Els usuaris GNU/Linux poden buscar fàcilment per Internet quins paquets del client i del servidor dhcp usant eines com yum, apt-get o wget, a part dels repositoris de programari usuals o del mateix Google.

Llista de paquets *rpm* que contenen exactament el text *dhcp*:

```
[root@portatil ~]# yum list dhcp
fedora                  100% |=========================| 2.1 kB    00:00
livna                   100% |=========================| 2.1 kB    00:00
updates                 100% |=========================| 2.3 kB    00:00
Installed Packages
dhcp.i386               12:3.0.5-42.fc7        installed
```

Llista de paquets que contenen la cadena *dhcp*:

```
[root@portatil ~]# yum list dhcp*
Installed Packages
dhcp.i386                               12:3.0.5-42.fc7        installed
dhcpv6_client.i386                      0.10-44.fc7            installed
Available Packages
dhcp-devel.i386                         12:3.0.5-42.fc7        updates
dhcp-forwarder.i386                     0.7-12.fc7             fedora
dhcp-forwarder-sysv.i386                0.7-12.fc7             fedora
dhcp-static.i386                        12:3.0.5-42.fc7        updates
dhcpv6.i386                             0.10-44.fc7            updates
```

Instal·lar el paquet *dhcp*:

```
# yum install dhcp
```

Fer la llista dels paquets *dhcp* instal·lats. Si el sistema ja els té instal·lats o volem comprovar-ho podem consultar els paquets instal·lats:

```
[root@portatil ~]# rpm -qa | grep dhcp
libdhcp6client-0.10-44.fc7
libdhcp4client-3.0.5-42.fc7
dhcp-3.0.5-42.fc7
dhcpv6_client-0.10-44.fc7
libdhcp-1.24-6.fc7
```

Obtenir informació del paquet del servei *dhcp* instal·lat:

```
[root@portatil ~]# rpm -qi dhcp
Name        : dhcp                         Relocations: (not relocatable)
Version     : 3.0.5                             Vendor: Fedora Project
```

```
Release     : 42.fc7                        Build Date: dl 12 nov 2007 17:37:56
     CET
Install Date: dc 23 gen 2008 19:14:18 CET      Build Host: xenbuilder4.fedora.
    phx.redhat.com
Group       : System Environment/Daemons    Source RPM: dhcp-3.0.5-42.fc7.src.
    rpm
Size        : 2162920                          License: ISC
Signature   : DSA/SHA1, dl 14 gen 2008 19:35:22 CET, Key ID b44269d04f2a6fd2
Packager    : Fedora Project
URL         : http://isc.org/products/DHCP/
Summary     : DHCP (Dynamic Host Configuration Protocol) server and relay agent
Description :
El DHCP (protocol de configuració dinàmica de màquines) és un protocol
que permet a dispositius individuals d'una xarxa IP obtenir la seva
informació de configuració de xarxa (adreça IP, màscara de subxarxa,
adreça de difusió, etc) d'un servidor DHCP. El propòsit del DHCP és
fer més senzilla l'administració d'una xarxa gran. El paquet dhcp
inclou el servei DHCP i l'agent de repetició de l'ISC.
Per usar DHCP a la vostra xarxa, instal·leu un servei DHCP (o agent
de repetició) i als clients executeu un dimoni client DHCP. El paquet
dhcp proporciona servei DHCP i l'agent de repetició de l'ISC.
```

1.3.3 Observar els components del programari instal·lat

Sovint passa que els usuaris instal·len programes als ordinadors però no saben quins fitxers han instal·lat ni on són. Un administrador curiós ha de fer un repàs a tot allò que s'ha afegit de nou al sistema.

Fer la llista dels components del paquet *dhcp*:

```
[root@portatil ~]# rpm -ql dhcp
/etc/dhcpd.conf
/etc/openldap/schema/dhcp.schema
/etc/rc.d/init.d/dhcpd
/etc/rc.d/init.d/dhcrelay
/etc/sysconfig/dhcpd
/etc/sysconfig/dhcrelay
/usr/bin/omshell
/usr/sbin/dhcpd
/usr/sbin/dhcrelay
/usr/share/doc/dhcp-3.0.5
... output suprimit ....
/usr/share/doc/dhcp-3.0.5/rfc951.txt
/usr/share/man/man1/omshell.1.gz
... output suprimit ...
/usr/share/man/man8/dhcrelay.8.gz
/var/lib/dhcpd
/var/lib/dhcpd/dhcpd.leases
```

En funció del directori on s'ubiquen els fitxers podem intuir si són executables, de configuració o de documentació. També podem mirar de filtrar la sortida en cada cas:

Fitxers de configuració (usualment en el directori */etc*):

```
[root@portatil ~]# rpm -qc dhcp
/etc/dhcpd.conf
/etc/openldap/schema/dhcp.schema
/etc/sysconfig/dhcpd
/etc/sysconfig/dhcrelay
```

```
/var/lib/dhcpd/dhcpd.leases

[root@portatil ~]# rpm -ql dhcp | grep etc
/etc/dhcpd.conf
/etc/openldap/schema/dhcp.schema
/etc/rc.d/init.d/dhcpd
/etc/rc.d/init.d/dhcrelay
/etc/sysconfig/dhcpd
/etc/sysconfig/dhcrelay
```

Fitxers de documentació:

```
[root@portatil ~]# rpm -qd dhcp
/usr/share/doc/dhcp-3.0.5/IANA-arp-parameters
/usr/share/doc/dhcp-3.0.5/README
/usr/share/doc/dhcp-3.0.5/README.ldap
/usr/share/doc/dhcp-3.0.5/RELNOTES
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/3.0b1-lease-convert
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/dhcp.spec
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/dhcpd-conf-to-ldap
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/ms2isc/Registry.perlmodule
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/ms2isc/ms2isc.pl
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/ms2isc/readme.txt
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/sethostname.sh
/usr/share/doc/dhcp-3.0.5/__fedora_contrib/solaris.init
/usr/share/doc/dhcp-3.0.5/api+protocol
/usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample
/usr/share/doc/dhcp-3.0.5/draft-ietf-dhc-authentication-14.txt
/usr/share/doc/dhcp-3.0.5/draft-ietf-dhc-dhcp-dns-12.txt
/usr/share/doc/dhcp-3.0.5/draft-ietf-dhc-failover-07.txt
/usr/share/doc/dhcp-3.0.5/draft-ietf-dhc-ldap-schema-01.txt
/usr/share/doc/dhcp-3.0.5/rfc1542.txt
/usr/share/doc/dhcp-3.0.5/rfc2131.txt
/usr/share/doc/dhcp-3.0.5/rfc2132.txt
/usr/share/doc/dhcp-3.0.5/rfc2485.txt
/usr/share/doc/dhcp-3.0.5/rfc2489.txt
/usr/share/doc/dhcp-3.0.5/rfc951.txt
/usr/share/man/man1/omshell.1.gz
/usr/share/man/man5/dhcp-eval.5.gz
/usr/share/man/man5/dhcp-options.5.gz
/usr/share/man/man5/dhcpd.conf.5.gz
/usr/share/man/man5/dhcpd.leases.5.gz
/usr/share/man/man8/dhcpd.8.gz
/usr/share/man/man8/dhcrelay.8.gz
```

Podem mirar de filtrar quins són els executables tenint en compte que usualment estaran en un directori de nom *bin* o *sbin*:

```
[root@portatil ~]# rpm -ql dhcp  | grep bin
/usr/bin/omshell                                /sbin
/usr/sbin/dhcpd                           /usr/sbin
                                         /bin
/usr/sbin/dhcrelay                      /usr/bin
```

Arxius executables

En GNU/Linux els fitxers executables per l'administrador es troben usualment a i a . Els executables d'usuari normalment són a i .

En resum:

- Els fitxers de documentació es troben generalment a: */usr/share/doc* i a */usr/share/man*.
- Els fitxers de configuració es troben a: */etc*, i a */etc/sysconfig*.
- El dimoni del servei es troba a: */usr/sbin/dhcpd*.
- El fitxer de configuració del dimoni del servei és: */etc/dhcpd.conf*.
- El fitxer de govern del servei és: */etc/rc.d/init.d/dhcpd*.

Activar/desactivar el servei i nivells d'arrancada

Un cop feta la instal·lació física de l'aplicació servidor DHCP (els fitxers) cal posar-lo en marxa, engegar-lo, o comprovar que ja ho està.

Serveis de xarxa: stand alone o xinetd

En la majoria de serveis de xarxa el servidor pot funcionar com un servei per si mateix (*stand alone*) o pot estar configurat per funcionar dins d'un superservei de xarxa com per exemple *inetd* i *xinetd*. Si funciona en mode de servei propi és el servidor qui escolta les connexions entrants i les atén. Si s'executa dins del superservei de xarxa, aquest és qui detecta les connexions entrants i activa el dimoni del servei per tal que les atengui, un cop ateses el dimoni del servei acaba i torna a ser el superservidor de xarxa qui es queda escoltant.

En un primer moment molts serveis funcionaven en mode *stand alone*, però això es va mostrar poc eficient a nivell de recursos. Fixeu-vos que si es vol oferir n serveis de xarxa, cal tenir engegats els n servidors corresponents. Aquests ocupen memòria i CPU, tot i que no atenguin cap client. Estan engegats només "per si" un client els demana.

Un model més eficient és engegar un únic procés que fa la tasca de superservidor de xarxa (*inetd o xinetd*) i només quan es rep una petició d'un client, s'executa el servei apropiat.

Usualment els serveis que funcionen en mode xinetd disposen de fitxers de configuració lligats al xinetd. Els serveis que actuen indivudualment disposen dels fitxers de configuració propis al etc rc.dinit.d/.

Molt sovint és decisió de l'administrador configurar en quin mode de funcionament vol que treballi el servei. Per exemple el servei de pàgines web (*httpd*), el servei *ssh* i molts d'altres, funcionen individualment. Serveis més bàsics com *telnet*, *finger*, *tftp*, etc funcionen usualment dins del *xinetd*.

Primerament cal saber si el servidor instal·lat funciona en mode *stand-alone* o dins del superdimoni de xarxa *xinetd* o *initd*. Si existeixen fitxers de configuració dins del directori */etc/xinetd.d/<nom-servei>* es tracta d'un servei dins del *xinetd*. Si existeixen fitxers de configuració dins del directori */etc/rc.d/init.d/<nom-servei>* es tracta d'un servei *stand-alone*. Podem fer:

```
[root@portatil ~]# rpm –ql dhcp  | grep /etc
/etc/dhcpd.conf
/etc/openldap/schema/dhcp.schema
/etc/rc.d/init.d/dhcpd
/etc/rc.d/init.d/dhcrelay
/etc/sysconfig/dhcpd
/etc/sysconfig/dhcrelay
```

Com podem observar es tracta d'un servei *stand-alone*. També es pot consultar el tipus de servei amb l'ordre chkconfig i observar si surt en la llista d'un tipus o de l'altre:

```
[root@portatil ~]# chkconfig ––list | grep dhcpd
dhcpd       0:apagat        1:apagat        2:apagat        3:apagat        4:
    apagat        5:apagat    6:apagat
```

Per facilitar buscar els serveis "stand-alone" podem fer:

```
[root@portatil ~]# chkconfig ––list dhcpd
dhcpd       0:apagat        1:apagat        2:apagat        3:apagat        4:
    apagat        5:apagat    6:apagat
```

L'estat del servei pot ser en execució (*running*) o aturat (*stopped*). Es pot consultar l'estat amb l'opció *status* de l'ordre *service* o del mateix executable del servei:

```
Es pot saber l'estat del servei amb l'opció status de les ordres:
[root@portatil ~]# service dhcp  status
dhcp  està aturat
[root@portatil ~]# /etc/rc.d/init.d/dhcpd status
dhcp  està aturat
```

La gestió de l'estat d'un servei normalment inclou les opcions *start*, *stop*, *status*, *restart* i *reload*. Aquestes són les més usuals, però cada servei pot definir les que cregui oportunes. Aquests són exemples de gestió de l'estat del servei DHCP:

Es pot arrancar el servei amb l'opció start de les ordres:

```
[root@portatil ~]# service dhcpd  start
S'està iniciant el servei dhcpd:                    [ FET  ]
[root@portatil ~]# /etc/rc.d/init.d/dhcpd start
S'està iniciant el servei dhcpd:                    [ FET  ]
```

Es pot aturar el servei amb l'opció stop de les ordres:

```
[root@portatil ~]# service dhcpd  stop
S'està aturant el dhcpd:                            [ FET  ]
[root@portatil ~]# /etc/rc.d/init.d/dhcpd stop
S'està aturant el servei dhcpd:                     [ FET  ]
```

Es pot iniciar de nou el servei (recarregar) amb l'opció reload o restart de les ordres:

```
[root@portatil ~]# /etc/rc.d/init.d/dhcpd restart
S'està aturant el servei dhcpd:                     [Incorrecte]
S'està iniciant el servei dhcpd:                    [ FET  ]
[root@portatil ~]# service dhcpd  reload
S'està aturant el servei dhcpd:                     [ FET  ]
S'està iniciant el servei dhcpd:                    [ FET  ]
```

Per saber les opcions possibles d'un servei es pot fer el truc:

```
[root@portatil ~]# service dhcpd  patapum
Forma d'ús: /etc/init.d/dhcpd {start|stop|restart|condrestart|status}
[root@portatil ~]# /etc/rc.d/init.d/dhcpd pimpam
Forma d'ús: /etc/rc.d/init.d/dhcpd {start|stop|restart|condrestart|status}
```

Que un servei del sistema estigui engegat no significa que en arrancar de nou el sistema torni a estar engegat. Cal comprovar que sigui així o assegurar-se de definir-ho correctament. Els serveis (usualment dimonis executables) es poden configurar per arrancar automàticament en determinats nivells d'execució. Les màquines GNU/Linux tenen 7 nivells d'execució com es pot veure del fitxer */etc/inittab*:

```
[root@portatil ~]# head -20 /etc/inittab
... output suprimit ...
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
```

```
#   6 - reboot (Do NOT set initdefault to this)
# ... output suprimit ...
```

Per configurar a quins nivells es vol que s'executi un servei s'utilitza l'ordre chkconfig, que permet activar/desactivar el servei pels nivells indicats. L'exemple següent mostra l'ús d'aquesta ordre, dóna la llista de l'estat actual del servei DHCP en tots els nivells i el configura per executar-se en els nivells 3, 4 i 5:

```
[root@portatil ~]# chkconfig --help
chkconfig versió 1.3.34 - Copyright (C) 1997-2000 Red Hat, Inc.
Aquest programari es pot distribuir lliurement d'acord amb els termes de la Llicència
Pública General GNU.
forma d'ús:   chkconfig --list [nom]
         chkconfig --add <nom>
         chkconfig --del <nom>
         chkconfig --override <nom>
         chkconfig [--level <nivells>] <nom> <on|off|reset|resetpriorities>
[root@portatil ~]# chkconfig --list dhcpd
dhcpd           0:apagat        1:apagat        2:apagat        3:apagat        4:apagat        5:apagat    6:apagat
[root@portatil ~]# chkconfig --level 345 dhcpd on
[root@portatil ~]# chkconfig --list | grep dhcpd
dhcpd           0:apagat        1:apagat        2:apagat        3:engegat       4:engegat       5:engegat       6:apagat
```

Fixeu-vos que definir els nivells d'execució no significa que el servei estigui ara mateix engegat. Significa que quan arranqui el sistema de nou s'engegarà en els nivells corresponents. Podem ara estar al nivell 5 i tenir el servei aturat perquè encara no l'hem engegat. Per exemple:

```
[root@portatil ~]# runlevel
N 5
[root@portatil ~]# service dhcpd  status
dhcpd  està aturat
[root@portatil ~]# service dhcpd  start
S'està iniciant el servei dhcpd                    [ FET  ]
```

1.4 Comprovació del funcionament bàsic

Comprovació DHCP

L'administrador pot verificar el funcionament del servidor DHCP observant:

- l'estat del servei (on, off)
- el PID del servei (ha d'estar running).
- el registre de *logs*.
- monitorar el trànsit de xarxa amb una eina tipus *wireshark*.
- l'estat dels ports.

Comprovar que el servidor DHCP està en funcionament és un procés ben senzill, n'hi ha prou de comprovar que el servei està engegat. Això no vol dir, en cap cas, que el servei estigui funcionant correctament. Potser el servidor està engegat però no està correctament configurat. De fet, la configuració és la part realment important de l'administració d'un servei, i també del servei DHCP.

A part de comprovar l'estat del servei (amb l'opció *status*), l'administrador pot assegurar-se que el dimoni del servei està en execució buscant el seu PID (*process identifier* o indicador de número de procés). Una altra activitat a fer és monitorar el registre d'activitats del servei (els *logs*). Tot el trànsit DHCP és trànsit de xarxa TCP/IP; per tant també es pot observar l'estat dels ports i analitzar el trànsit que s'hi produeix.

Tot procés en el sistema té un identificador de procés. El PID dels serveis usualment es desen en el sistema de fitxers (al directori */var/run*) en forma de

fitxer que conté un valor numèric (en text) corresponent al PID del procés. Amb el servei en marxa sempre es pot observar el PID del servidor amb:

```
[root@portatil ~]# ps ax | grep dhcp
 3610 ?        Ss     0:00 /usr/sbin/dhcpd
[root@portatil ~]# service dhcpd status
dhcpd (pid 3610) s'està executant...
[root@portatil ~]# ll /var/run/dhcpd.pid
-rw-r--r-- 1 root root 5 29 jun 14:17 /var/run/dhcpd.pid
[root@portatil ~]# cat /var/run/dhcpd.pid
3610
```

Un cop iniciat el servei es crea un fitxer de bloqueig o *lock* amb el nom del servei per evitar iniciar-ne una altra instància. Els fitxers de *lock* usualment es troben a */var/lock* i són un simple fitxer de text buit on la seva pròpia existència ja marca que el servei està en marxa. En parar el servei el fitxer s'elimina. Podem observar això fent:

```
[root@portatil ~]# cat /var/lock/subsys/dhcpd
[root@portatil ~]# ll /var/lock/subsys/dhcpd
-rw-r--r-- 1 root root 0  1 jun 18:26 /var/lock/subsys/dhhcpd
```

Tots els serveis del sistema normalment es monitoren anotant en fitxers de text un registre de totes les accions que realitzen, són els fitxers coneguts com a fitxers de *log*. Tant es pot utilitzar un fitxer genèric pel sistema com un fitxer independent per a un servei determinat. El servidor DHCP utilitza el fitxer de monitoratge estàndard */var/log/messages*. En aquest fitxer s'enregistra cada cop que el servei s'engega i s'atura.

```
[root@portatil ~]# cat /var/log/messages | grep dhcp
Jun 29 14:16:53 portatil yum: Installed: dhcp - 12:3.0.5-42.fc7.i386
Jun 29 14:17:33 portatil dhcpd: Internet Systems Consortium DHCP Server V3.0.5-
    RedHat
Jun 29 14:17:33 portatil dhcpd: Copyright 2004-2006 Internet Systems Consortium
    .
Jun 29 14:17:33 portatil dhcpd: All rights reserved.
Jun 29 14:17:33 portatil dhcpd: For info, please visit http://www.isc.org/sw/
    dhcp/
Jun 29 14:17:33 portatil dhcpd: WARNING: Host declarations are global.  They
    are not limited
to the scope you declared them in.
Jun 29 14:17:33 portatil dhcpd: Wrote 0 deleted host decls to leases file.
Jun 29 14:17:33 portatil dhcpd: Wrote 0 new dynamic host decls to leases file.
Jun 29 14:17:33 portatil dhcpd: Wrote 0 leases to leases file.
Jun 29 14:17:33 portatil dhcpd: Listening on LPF/eth0/00:17:31:15:80:7e
    /192.168.1/24
Jun 29 14:17:33 portatil dhcpd: Sending on   LPF/eth0/00:17:31:15:80:7e
    /192.168.1/24
```

El servei DHCP desa la informació de registre de les concessions que efectua en un fitxer de *leasses*. Això li permet seguir la pista de les adreces IP que ha concedit i ser coherent entre diverses arrancades del mateix servidor. Es pot observar aquest fitxer a */var/lib/dhcpd/dhcpd.leases*:

```
[root@portatil ~]# ll /var/lib/dhcpd/dhcpd.leases
-rw-r--r-- 1 root root 473 29 jun 14:17 /var/lib/dhcpd/dhcpd.leases

[root@portatil ~]# cat /var/lib/dhcpd/dhcpd.leases
# All times in this file are in UTC (GMT), not your local timezone.   This is
# not a bug, so please don't ask about it.   There is no portable way to
```

```
# store leases in the local timezone, so please don't request this as a
# feature.   If this is inconvenient or confusing to you, we sincerely
# apologize.   Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-V3.0.5-RedHat
lease 192.168.1.254 {
  starts 0 2008/06/29 16:03:41;
  ends 1 2008/06/30 04:03:41;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:b2:8b:ec;
  client-hostname "box";
}

lease 192.168.1.254 {
  starts 0 2008/06/29 16:19:30;
  ends 1 2008/06/30 04:19:30;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:b2:8b:ec;
  client-hostname "box";
}
```

Sovint l'administrador vol comprovar que els ports que utilitza el protocol DHCP estan oberts. En GNU/Linux es pot fer una llista fàcilment dels serveis associats a cada port mitjançant el fitxer */etc/services*. Algunes utilitats com nmap permeten detectar els ports oberts. Per exemple podem fer:

```
Llista dels ports que inclouen alguna referència DHCP:
[root@portatil ~]# cat /etc/services  | grep DHCP
bootpc          68/tcp          dhcpc           # BOOTP client
bootpc          68/udp          dhcpc
dhcpv6-client   546/tcp
dhcpv6-client   546/udp
dhcpv6-server   547/tcp
dhcpv6-server   547/udp
dhcp-failover   647/tcp                         # DHCP Failover
dhcp-failover   647/udp                         # DHCP Failover
dhcp-failover2  847/tcp                         # dhcp-failover 2
dhcp-failover2  847/udp                         # dhcp-failover 2
qip-qdhcp       2490/tcp                        # qip_qdhcp
qip-qdhcp       2490/udp                        # qip_qdhcp
```

Si ens hi fixem veurem que de fet el protocol DHCP és anomenat BOOTP. Podem fer la llista de les entrades que corresponen a aquest protocol:

```
[root@portatil ~]# cat /etc/services | grep bootp
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp          dhcpc           # BOOTP client
bootpc          68/udp          dhcpc
nuts_bootp      4133/tcp                        # NUTS Bootp Server
nuts_bootp      4133/udp                        # NUTS Bootp Server
```

Es pot observar que el port usat pel client DHCP és el port 68 (bootpc) i el port usat pel servidor és el port 67 (bootps).

1.5 Configuració DHCP

Per configurar el servei DHCP primer cal saber observar i manipular la configuració de xarxa existent, i això consisteix a saber:

- Observar/fer la llista de la configuració de xarxa actual.
- Comprovar l'estat del servei de xarxa.
- Activar/desactivar el servei de xarxa.
- Monitorar el servei i el procés del servidor.

Exemples de configuració

En GNU/Linux és molt usual que el paquet que proporciona un servei inclogui un fitxer d'exemple de configuració. El servei DHCP incorpora el fitxer d'exemple *dhcpd..conf* i la pàgina de manual del mateix nom.

Les tasques principals per configurar un servidor DHCP són les següents:

- Instal·lar el programari del servidor DHCP.
- Activar/desactivar el servei de DHCP.
- Observar/fer la llista de la configuració actual del servidor DHCP.
- Modificar la configuració del servidor DHCP.
- Monitorar els *logs* del servei DHCP i els fitxers de registre de les concessions (*leases*).

Per tant, abans d'endinsar-nos en la configuració del servei és molt útil observar una configuració ja existent. Un exemple de fitxer de configuració del servei DHCP és el que es mostra a continuació:

```
# a) opcions globals del servidor DHCP (usuals)
ddns-update-style interim;
ignore client-updates;
# b) definició de la xarxa a la qual s'ofereix el servei DHCP
subnet 192.168.0.0 netmask 255.255.255.0 {
   # opcions genèriques per a tots els equips de la xarxa
   option routers      192.168.0.1;
   option subnet-mask     255.255.255.0;
   option domain-name     "domain.org";
   option domain-name-servers  192.168.1.1;
   # definició de l'interval d'ips dinàmiques a usar
   # i dels temps de les concessions
   range dynamic-bootp 192.168.0.128 192.168.0.254;
   default-lease-time 21600;
   max-lease-time 43200;
  # c) Opcions d'equips individuals
  # el servidor ns obté sempre una adreça fixa basada en MAC
  host ns {
   next-server marvin.redhat.com;
   hardware ethernet 12:34:56:78:AB:CD;
   fixed-address 207.175.42.254;
  }
}
```

En aquest fitxer de configuració es pot veure que hi ha tres àmbits diferents de definició:

1. **Opcions globals en el servidor DHCP.** Són opcions que indiquen al servidor la seva manera d'actuar. També són opcions generals que cal aplicar a totes les concessions que es realitzin, independentment de la xarxa o equip que siguin.

2. **Definicions i opcions de xarxa.** Es defineixen tantes xarxes diferents com subxarxes ha d'atendre el servidor. Cada definició de subxarxa consta de la IP de la xarxa i la màscara corresponent. Entre claus s'indiquen totes les opcions específiques per a les concessions de les adreces IP corresponents a aquesta subxarxa. Una opció característica és indicar l'interval (o *pool*) d'adreces dinàmiques a usar, la porta d'enllaç predeterminada, el servidor de noms, etc.

3. **Opcions d'equips individuals.** Dins d'una subxarxa es poden definir opcions per a equips individuals. Cal identificar els equips per la seva adreça MAC i, entre claus, indicar les opcions que els són específiques. Això permet assignar adreces fixes dinàmicament (equivalent al protocol BOOTP) usant les opcions de maquinari Ethernet i *fixed-address*.

Les opcions globals de configuració DHCP es poden redefinir amb valors diferents dins d'un bloc de xarxa concret. Dins d'un equip també es poden definir opcions amb valors diferents als definits per la xarxa o globalment. Tal com passa en els llenguatges de programació preval el valor més intern, el de *host* sobre el de xarxa i el de xarxa per sobre del global.

1.5.1 Configuració bàsica

Per fer funcionar el servidor DHCP cal configurar-lo prèviament. Per poder arrancar li cal saber a quina xarxa donarà servei i quin és l'interval d'adreces IP que pot usar dinàmicament per a les concessions als clients.

El paquet DHCP conté un fitxer d'exemple al directori */user/share/-doc/dhcp*/dhcpd.conf.sample*. Aquest fitxer es pot copiar a */etc/dhcpd.conf* i passarà a ser la configuració bàsica del servidor DHCP. Podem veure el seu contingut fent:

```
[root@portatil ~]# ll /usr/share/doc/dhcp-3.0.5/dhcpd.conf.sample
-rw-r--r-- 1 root root 852 12 nov  2007 /usr/share/doc/dhcp-3.0.5/dhcpd.conf.
    sample
[root@portatil ~]# cat /etc/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#
#a) opcions globals
ddns-update-style interim;
ignore client-updates;

#b) opcions genèriques per a una subxarxa concreta
subnet 192.168.1.0 netmask 255.255.255.0 {
        # --- default gateway
        option routers                  192.168.1.1;
```

```
        option subnet-mask              255.255.255.0;
        option nis-domain               "domain.org";
        option domain-name              "domain.org";
        option domain-name-servers      192.168.1.1;
        option time-offset              -18000; # Eastern Standard Time
        range dynamic-bootp 192.168.1.128 192.168.1.254;
        default-lease-time 21600;
        max-lease-time 43200;

        #c) we want the nameserver to appear at a fixed address
        host ns {
                next-server marvin.redhat.com;
                hardware ethernet 12:34:56:78:AB:CD;
                fixed-address 207.175.42.254;
        }
}
```

En la configuració per defecte es poden analitzar els diversos elements que es configuren:

- Opcions globals: indiquen al servidor que ignori les actualitzacions dels clients i el tipus de DDNS a usar (actualitzacions dinàmiques de DNS).
- Definició de subxarxa: cal definir tants blocs de subxarxa com subxarxes atengui el servidor DHCP.
- Opcions genèriques de subxarxa: per a una subxarxa es poden indicar opcions genèriques per als equips d'aquesta subxarxa. Evidentment poden diferir de les opcions d'altres subxarxes.
- Les principals opcions de xarxa a descriure són el router, la màscara de xarxa, el domini, etc.
- Les principals opcions a descriure del servei DHCP són definir l'interval d'adreces IP dinàmiques a usar pel servidor i el temps màxim de concessió d'aquestes adreces IP.
- Per tal que un *host* determinat tingui sempre la mateixa adreça IP es poden fer entrades individualitzades per a *hosts* concrets. Els *hosts* s'identifiquen per la seva adreça MAC.
- A un *host* concret (descrit en una entrada *host*) se li poden aplicar opcions individualitzades, com per exemple definir el seu nom. Les opcions individuals prevalen sobre les genèriques.

1.5.2 Configuració avançada

El protocol DHCP permet configuracions d'una certa complexitat. Podeu consultar la documentació de DHCP i les pàgines de manual del dimoni *dhcpd* i del fitxer de configuració *dhcpd.conf*.

Les principals característiques que s'hi descriuen són l'agrupació d'entrades en grups i classes i la possibilitat que el DHCP es comuniqui amb el DNS (actualitzacions DDNS) per crear entrades DNS quan un equip rep una configuració DHCP.

Un exemple de configuració amb opcions més avançades:

```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.1.0 netmask 255.255.255.0 {
# --- default gateway

option routers 192.168.1.1;
option subnet-mask 255.255.255.0;

option nis-domain "domain.org";
option domain-name "domain.org";
option domain-name-servers192.168.1.1;
option time-offset-18000;# Eastern Standard Time

range dynamic-bootp 192.168.1.128 192.168.1.254;
default-lease-time 21600;
max-lease-time 43200;

  # we want the nameserver to appear at a fixed address
  host ns {
     next-server marvin.redhat.com;
     hardware ethernet 12:34:56:78:AB:CD;
     fixed-address 207.175.42.254;
  }

   group {
      filename "Xncd19r";
      next-server ncd-booter;

      host ncd1 { hardware ethernet 0:c0:c3:49:2b:57; }
      host ncd4 { hardware ethernet 0:c0:c3:80:fc:32; }
      host ncd8 { hardware ethernet 0:c0:c3:22:46:81; }
   }
}

subnet 10.0.0.0 netmask 255.255.255.0 {
         option routers 10.0.0.254;
         # Unknown clients get this pool.
         pool {
           option domain-name-servers bogus.example.com;
           max-lease-time 300;
           range 10.0.0.200 10.0.0.253;
           allow unknown-clients;
         }

         # Known clients get this pool.
         pool {
           option domain-name-servers ns1.example.com, ns2.example.com;
           max-lease-time 28800;
           range 10.0.0.5 10.0.0.199;
           deny unknown-clients;
         }
}
```

1.5.3 Base de dades de concessions fetes pel servidor

El servidor desa en una base de dades local (de fet són fitxers de text) les concessions (*leases*) que realitza. D'aquesta manera en pot seguir la pista en tot moment. Generalment, les té a memòria (per permetre un accés més ràpid), però en manté una còpia a disc. Si, per exemple, el sistema o el servei es reinicia, pot

saber quines són les concessions que encara estan actives (i per tant quines adreces IP no té disponibles).

Usualment el fitxer de concessions és a */var/lib/dhcpd*. Podem veure'n el contingut fent:

```
[root@portatil ~]# cat /var/lib/dhcpd/dhcpd.leases
# All times in this file are in UTC (GMT), not your local timezone. This is
# not a bug, so please don't ask about it. There is no portable way to
# store leases in the local timezone, so please don't request this as a
# feature.   If this is inconvenient or confusing to you, we sincerely
# apologize.   Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-V3.0.5-RedHat

lease 192.168.1.254 {
  starts 0 2008/06/29 16:03:41;
  ends 1 2008/06/30 04:03:41;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:b2:8b:ec;
  client-hostname "box";
}
lease 192.168.1.254 {
  starts 0 2008/06/29 16:19:30;
  ends 1 2008/06/30 04:19:30;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:b2:8b:ec;
  client-hostname "box";
}
lease 192.168.1.253 {
  starts 0 2008/06/29 16:27:16;
  ends 0 2008/06/29 22:27:16;
  binding state active;
  next binding state free;
  hardware ethernet 08:00:27:8e:72:de;
}
```

1.5.4 Opcions de configuració del servidor i àmbit d'aplicació

Les opcions de configuració DHCP són múltiples i comprenen molts àmbits. Algunes permeten compatibilitat amb sistemes antics, d'altres amb altres tipus de xarxes, etc. No és imaginable que un administrador de xarxes les conegui totes a fons. Normalment farà ús d'un conjunt reduït d'opcions que serà més que suficient per administrar la majoria de xarxes.

La configuració DHCP es pot definir tant al client com al servidor, tot i que usualment es farà en el servidor. Entenem com a tasca principal la de configurar un servidor per tal de proporcionar les opcions apropiades a cada subxarxa. De totes maneres, però, un client també pot disposar d'un fitxer de configuració on es defineixen quins són els seus requeriments i com ha de realitzar el diàleg amb el servidor. Per exemple es defineixen quines opcions ha de sol·licitar, valors per defecte a determinades opcions (per si el servidor no en proporciona un valor). El client també pot definir informació que proporcionarà al servidor per tal que aquest prengui decisions dinàmicament.

Podeu fer un repàs més extens de la configuració del servidor consultant la secció d'annexos del web d'aquest mòdul.

Caldrà, doncs, entendre quins són els àmbits (scope) de definició de sentències i opcions, com s'agrupen les subxarxes i els *hosts*, quines són les opcions globals, com es realitzen definicions condicionals i molts altres detalls.

Àmbit de definició

Els clients es poden agrupar en diversos àmbits per tal de definir les opcions que han de rebre. El mateix servidor DHCP pot actuar de manera diferent segons sigui l'àmbit de definició.

Alguns dels conceptes a tractar són:

- *Subnets*.
- Període de concessió.
- Adreces fixes o reservades: identificació de *host*.
- PXE: Protocol d'arrencada via xarxa.
- Àmbit d'aplicació.
- *Pool*.

Subnets

El servei DHCP permet fer assignació dinàmica d'adreces IP per diferents subxarxes sense que es requereixi un servidor específic per a cada subxarxa. Fins i tot el servei es pot oferir a xarxes "llunyanes", és a dir, que han de creuar almenys un router per accedir al servidor. En aquest cas es parla del concepte DHCP Relaying.

Per oferir el servei a una subxarxa cal conèixer l'adreça de la xarxa i la seva màscara corresponent. Per cada subxarxa el servidor pot disposar d'un o més intervals d'adreces dinàmiques de les quals obtindrà l'adreça a concedir al client. Uns exemples bàsics poden ser:

```
# subxarxa amb un interval d'adreces dinàmiques:
subnet 239.252.197.0 netmask 255.255.255.0 {
    range 239.252.197.10 239.252.197.250;
}

# subxarxa amb dos intervals d'adreces dinàmiques:
subnet 239.252.197.0 netmask 255.255.255.0 {
    range 239.252.197.10 239.252.197.107;
    range 239.252.197.113 239.252.197.250;
}
```

Període de concessió

Les concessions d'adreces s'efectuen per períodes de temps determinats. Un cop exhaurit aquest temps, cal procedir a renegociar la concessió (que es pot renovar o cancel·lar). De fet les concessions es poden efectuar per períodes des de zero

segons a un temps infinit. El temps apropiat en cada cas dependrà del tipus de client i servei que s'ofereix; és diferent un client wireless a l'aeroport que l'estació de treball del supercap executiu de la corporació.

Des del punt de vista de la configuració del servidor es poden establir dos tipus de temps, el *default-lease-time* i el *max-lease-time*. El primer permet definir el temps màxim per defecte que es concedeix als clients quan aquests no han sol·licitat cap període de temps concret. El segon estableix el temps màxim de concessió en aquesta subxarxa. Podem veure un exemple d'aquestes opcions:

```
# el temps màxim d'una concessió són 7200 segons, a cap client se li pot
    concedir més temps. Si el client no ha demanat cap temps concret, se li
    assigna una concessió de 600 segons.

subnet 239.252.197.0 netmask 255.255.255.0 {
       range 239.252.197.10 239.252.197.107;
       default-lease-time 600;
       max-lease-time 7200;
}
```

Els períodes de concessió es poden definir globalment per a totes les subxarxes a les quals es proporciona servei, però és més usual fer-ho per cada subxarxa quan aquestes tenen requeriments diferents.

Adreces reservades

A part de l'assignació dinàmica d'adreces d'interval el servidor DHCP també pot fer assignacions dinàmiques fixes. És a dir, assignar a un equip sempre la mateixa dreça IP. En aquest cas parlem d'adreça reservada. L'equip 'consumeix' aquesta adreça tant si està engegat com si està apagat. Per poder assignar a un *host* concret sempre la mateixa adreça cal identificar-lo de manera única i inequívoca. Això es pot fer mitjançant la seva adreça MAC. Un exemple pot ser:

```
host iocserver {
     hardware ethernet 08:00:2b:4c:59:23;
     fixed-address 239.252.197.9;
}
```

Es pot observar que l'opció hardware permet identificar el *host* iocserver a través de la seva adreça MAC.

PXE: protocol d'arrencada via xarxa

Molts sistemes i targetes de xarxa permeten arrencar clients de xarxa 'tontos'. És a dir, equips que arranquen sense disposar de sistema operatiu i que el carreguen via xarxa. PXE és un protocol àmpliament utilitzar per a aquest fi. El client rep via DHCP el nom d'un fitxer que caldrà que descarregui via TFTP. Aquest fitxer acostuma a ser el sistema operatiu o el programari que ha de carregar per inicialitzar-se. L'opció DHCP que comunica el nom del fitxer a descarregar és l'opció *filename*.

```
host iocserver {
     hardware ethernet 08:00:2b:4c:59:23;
     fixed-address 239.252.197.9;
     filename "/tftpboot/kernel_ioc.boot";
}
```

En aquest exemple s'indica el fitxer *kernel_ioc.boot* per descarregar per TFTP si s'utilitza un protocol d'arrencada via xarxa com el PXE.

Opcions generals

El servei DHCP proveeix el client no només de l'adreça IP, màscara i gateway, sinó de molts altres paràmetres de configuració de xarxa. Alguns d'aquests paràmetres són molt usuals (DNS, *routers*, etc) i d'altres molt específics (rars fins i tot).

Tots aquests paràmetres es passen al client en forma d'*options*. Les opcions poden ser d'àmbit general (per a totes les subxarxes), d'àmbit per a una subxarxa i d'àmbit concret d'un sol *host*. L'ordre de precedència és de l'àmbit més concret al més general. Un exemple seria:

Per saber més de les opcions específiques es pot consultar la pàgina del manual dhcp-options(5) en un sistema GNU/Linux.

```
subnet 239.252.197.0 netmask 255.255.255.0 {
     range 239.252.197.10 239.252.197.250;
     default-lease-time 600
     max-lease-time 7200;
     option subnet-mask 255.255.255.0;
     option broadcast-address 239.252.197.255;
     option routers 239.252.197.1;
     option domain-name-servers 239.252.197.2, 239.252.197.3;
     option domain-name "isc.org";

     host iocserver {
         hardware ethernet 08:00:2b:4c:59:23;
         fixed-address 239.252.197.9;
         filename "/tftpboot/kernel_ioc.boot";
         option domain-name-servers 192.5.5.1;
         option domain-name "vix.com";
 }
}
```

En l'exemple anterior s'observa que es defineixen les opcions *subnetmask*, *broadcast-address*, *routers*, *domain-name-servers* i *domain-name* per a tots els equips de la subxarxa 239.252.197.0/24. Hi ha dues opcions, però, que es redefineixen amb valors diferents per al *host iocserver*. Són les opcions *domain-name-servers* i *domain-name*.

Àmbit d'aplicació: subnet

Les topologies de xarxa s'estructuren usualment en subxarxes, el mateix passa en la configuració del servidor DHCP. Usualment els clients s'agrupen en la subxarxa a la qual pertanyen. S'utilitza la sentència *subnet* per definir les opcions d'una subxarxa concreta. Si els clients de la subxarxa han de rebre adreces dinàmiques d'interval, cal com a mínim una sentència *range*.

De vegades convé agrupar subxarxes diferents en un sol bloc d'opcions de configuració. Això passa per exemple quan una sola xarxa física que es vol

tractar de la mateixa manera tota ella està dividida en dues subxarxes lògiques. La sentència *shared-network* s'utilitza en aquestes ocasions.

> Les sentències d'àmbit d'aplicació més usuals són subnet i host, que permeten identificar una subxarxa i un host concret respectivament.
> Les subxarxes es poden agrupar en shared-network, els clients es poden agrupar usant la sentència **group**.
> Les opcions es poden definir en funció de determinats requisits que compleixi el client mitjançant la sentència **class** i les **declaracions condicionals**.

Quan els clients han de rebre una adreça IP dinàmica fixa, és a dir, han de rebre sempre la mateixa IP reservada, cal especificar la sentència *host*. Si es vol que tots els clients siguin únicament equips identificats cal fer una sentència *host* per a cada equip.

A vegades es vol establir configuracions que afecten equips i xarxes que no tenen una agrupació en forma de subxarxa clara. En aquests casos es pot optar per utilitzar la sentència *group*. Amb aquesta sentència es poden fer agrupacions d'elements diversos als quals se'ls assignen opcions comunes.

Una altra situació que es pot produir és voler agrupar clients segons determinades condicions que acompleixi el client (no únicament segons la seva MAC). La sentència class permet agrupar opcions segons la informació que envia el mateix client. A més a més, es poden utilitzar declaracions condicionals, és a dir, segons es compleixi o no una determinada condició aplicar unes declaracions o unes altres.

El següent és un exemple de l'esquema bàsic d'un fitxer de configuració dhcp:

```
#exemple extret de la pàgina de configuració del dhcpd.conf(5)
       global parameters...
  option domain-name "ioc.org";
       option domain-name-servers ns1.ioc.org, ns2.ioc.org;
       max-lease-time 7200;
       default-lease-time 600;
       subnet 204.254.239.0 netmask 255.255.255.224 {
         subnet-specific parameters...
         option routers 204.254.239.1;
         range 204.254.239.10 204.254.239.30;
       }
       subnet 204.254.239.32 netmask 255.255.255.224 {
         subnet-specific parameters...
         option routers 204.254.239.33;
         range 204.254.239.42 204.254.239.62;
       }
       subnet 204.254.239.64 netmask 255.255.255.224 {
         subnet-specific parameters...
         option routers 204.254.239.65;
         range 204.254.239.74 204.254.239.90;
         host grouxo {
           host-specific parameters...
           hardware ethernet 08:00:2b:4c:59:23;
           filename "/tftpboot/grouxo.img";
           fixed-address 204.254.239.91;
         }
       }
       group {
```

```
        group-specific parameters...
        option domain-name "marxbrothers.ioc.org";
        max-lease-time 120;
        host zappo.test.isc.org {
          host-specific parameters...
        }
        host harpo.test.isc.org {
          host-specific parameters...
        }
      }
```

En aquest exemple podem observar que:

- es defineixen opcions globals per a totes les xarxes, com per exemple el nom del domini, els servidors de noms (DNS) i els temps de les concessions.
- per a cada subxarxa es defineixen opcions específiques com poden ser per exemple l'interval d'adreces disponibles, el router de la subxarxa, etc. També es pot observar que en la subxarxa 204.254.239.64/27 es defineixen opcions particulars per al *host grouxo*. Concretament obté una adreça IP de l'interval fixa i un fitxer d'inicialització via TFTP.
- finalment l'exemple mostra com es poden definir opcions per agrupacions amb la sentència *group*. Els *hosts harpo* i *zappo* estan agrupats i reben un nom de domini particular, diferent al definit en les opcions globals. També se'ls defineix un valor de temps màxim de concessió diferent al global.

Tot seguit es mostra un altre exemple d'estructura de configuració DHCP on s'agrupen *hosts* en grups d'afinitats:

```
#exemple extret de la pàgina de configuració del dhcpd.conf(5)
   group {
         filename "osOficines";
         next-server ncd-booter;
         host ncd1 { hardware ethernet 0:c0:c3:49:2b:57; }
         host ncd4 { hardware ethernet 0:c0:c3:80:fc:32; }
         host ncd8 { hardware ethernet 0:c0:c3:22:46:81; }
   }
   group {
         filename "osProfessorat";
         next-server ncd-booter;
         host ncd2 { hardware ethernet 0:c0:c3:88:2d:81; }
         host ncd3 { hardware ethernet 0:c0:c3:00:14:11; }
  }
  group {
         filename "osAlumnes";
         next-server alumni-booter;
         host ncd1 { hardware ethernet 0:c0:c3:11:90:23; }
         host ncd4 { hardware ethernet 0:c0:c3:91:a7:8; }
         ...output suprimt ...
         host ncd80 { hardware ethernet 0:c0:c3:cc:a:8f; }
 }
```

En aquest exemple s'han creat tres grups diferents segons si es tracta d'equips de les oficines, del professorat o dels alumnes. S'ha indicat individualment equip per equip mitjançant la sentència *host*. A cada grup se'ls assigna les opcions que els són comunes (en lloc de fer-ho repetidament dins de cada *host*). En aquest cas representa que els equips són terminals "ximples" que carreguen el sistema

operatiu des de la xarxa. Cada grup carrega un sistema operatiu diferent. Oficines i professors el descarreguen del mateix servidor, mentre que els alumnes ho fan d'un altre anomenat *alumni-booter*.

Existeix un altre mecanisme per agrupar declaracions anomenat *pool* i que permet fer agrupacions d'adreces. Cada agrupació d'adreces o *pool* pot ser tractat de manera diferent. Es poden definir diferents pool dins d'una mateixa subxarxa (*subnet*). Possiblement, la millor manera d'entendre com funciona és mitjançant un exemple:

```
#exemple extret de la pàgina de configuració del dhcpd.conf(5)
    subnet 10.0.0.0 netmask 255.255.255.0 {
         option routers 10.0.0.254;
         # Unknown clients get this pool.
         pool {
           option domain-name-servers noexisteixo.com;
           max-lease-time 300;
           range 10.0.0.200 10.0.0.253;
           allow unknown-clients;
   }
# Known clients get this pool.
         pool {
           option domain-name-servers ns1.example.com, ns2.example.com;
           max-lease-time 28800;
           range 10.0.0.5 10.0.0.199;
           deny unknown-clients;
         }
    }
```

En aquest exemple la subxarxa 10.0.0.0/8 es divideix en dos *pool* d'adreces per ser tractats de manera diferent. En el primer s'assignen dinàmicament adreces IP de la 200 a la 253 i s'assigna també un servidor de noms inexistent. Seran clients d'aquest *pool* els equips que siguin no identificats (*unknown-clients*). En el segon pool s'accepten només clients coneguts, als quals s'assignen adreces IP de la 5 a la 199 i dos servidors de noms existents.

D'aquest exemple podem observar que la sentència *pool* permet establir polítiques d'accés (*permit lists*) definint quins clients poden accedir al *pool* i quins no. El tractament de les polítiques d'accés queda fora de l'abast d'aquest llibre, però és molt intuïtiu. La clàusula *allow* indica els clients amb accés i la clàusula *deny* els clients que no podran accedir al *pool*.

Polítiques d'accés

Per saber més de les polítiques d'accés es pot consultar la pàgina de manual del dhcpd.conf(5) en un sistema GNU/Linux.

Class

Una de les eines més potents per agrupar clients és la sentència class. Els clients es poden tractar de manera diferent segons a quina classe pertanyin. Per determinar la classe del client es poden usar les dades que el mateix client proporciona. S'utilitzen sentències condicionals i tractament d'expressions (*conditional, match statement*) que s'avaluen per determinar si el client pertany o no a una classe. Per exemple es pot crear una classe amb tots aquells clients que contenen l'identificador "IOC":

Les classes es poden subdividir en subclasses que corresponen a valors específics que satisfan l'expressió de la classe. El tractament de les classes, condicions i

Classes

Per saber més de les classes es pot consultar la pàgina de manual del dhcpd.conf(5) en un sistema GNU/Linux. Per saber com tractar i avaluar expressions es pot consultar el manual dhcp-eval(5) en un sistema GNU/Linux. Per saber més de les opcions específiques es pot consultar el manual dhcp-options(5) en un sistema GNU/Linux.

expressions queda fora de l'abast d'aquest material. Per conèixer bé les classes cal conèixer el tractament d'expressions i les opcions específiques del DHCP.

1.5.5 Sentències i opcions de configuració

El servei DHCP es pot configurar amb multitud de sentències que es poden repassar de l'RFC 2131 i de la pàgina de manual dhcpd.conf(5). Examinem tot seguit alguna d'aquestes sentències.

Declaracions

- **Include:** include "filename"; Permet carregar el fitxer indicat i processar-lo com a fitxer de configuració. És una tècnica usual per dividir la configuració en mòduls.
- **Shared-network:** shared-network name { [parameters] [declarations] } Permet agrupar diverses subxarxes (subnet) en una mateixa declaració. S'utilitza quan una mateixa xarxa física es compon de diverses subxarxes lògiques.
- **Subnet:** subnet subnet-number netmask netmask { [parameters] [declarations] } Permet definir opcions per a una subxarxa concreta. És la sentència més usual en les definicions de configuració del servidor dhcp.
- **Range:** range [dynamic-bootp] low-address [high-address]; Indica l'interval d'adreces dinàmiques disponibles per assignar. El servidor DHCP extreu les adreces dinàmiques d'aquest interval d'adreces.
- **Host:** host hostname { [parameters] [declarations] } Proporciona un àmbit de definició per a un equip concret. Les opcions que es defineixen dins d'una sentència *host* afecten únicament l'equip indicat. Es requereix una sentència *host* per poder fer assignacions dinàmiques fixes (assignar sempre la mateixa adreça IP basant-se en l'adreça MAC).
- **Group:** group { [parameters] [declarations] } S'utilitza per agrupar declaracions de manera que les opcions definides afectin el grup d'elements que conté. Aquests poden ser *shared-networks*, *subnets*, *hosts* i fins i tot altres grups.

Paràmetres

Tot seguit es mostren alguns dels paràmetres que es poden definir en el servidor DHCP.

- **authoritative** authoritative; not authoritative; Indica si les respostes del servidor DHCP són autoritatives o no. Si ho són el servidor revocarà les adreces IP que ell no ha concedit.

- **ddns-update-style** ddns-update-style style; Pot ser *ad-hoc* o *interim* o *none*. Indica el tipus de *dynamyc DNS* que s'utilitza.
- **default-lease-time** default-lease-time time; Indica el temps per defecte de les concessions. És el temps que es concedeix quan el client no requereix un període de temps concret.
- **fixed-address** fixed-address address [, address ...]; Permet assignar una o més adreces IP a un client.
- **hardware** hardware hardware-type hardware-address; Indica el tipus i el valor de l'adreça Mac d'un client. Aquest mecanisme és necessari per poder identificar un client de manera única, per exemple en una clàusula *host*.
- **max-lease-time** max-lease-time time; Indica el temps màxim per a les concessions en segons.
- **min-lease-time** min-lease-time time; Indica el temps mínim que han de durar les concessions.
- **next-server** next-server server-name; Indica el nom o l'adreça IP del servidor TFTP del qual el client s'ha de descarregar el fitxer d'inicialització (o *boot*) indicat pel paràmetre *filename*. Si no s'indica s'utilitza el propi servidor DHCP.

Opcions de configuració

Els clients DHCP reben del servidor la configuració de xarxa. Usualment parlem de l'adreça IP i la màscara, però de fet poden rebre gran quantitat

de paràmetres de configuració de xarxa i informació sobre diversos serveis de xarxa disponibles. El client per la seva part pot sol·licitar paràmetres concrets al servidor. L'administrador de xarxa, quan configura el servei DHCP, no ha d'especificar totes les opcions possibles (de fet són moltíssimes), sinó només aquelles que siguin necessàries per a cada client. Algunes opcions prenen valors per defecte i no cal especificar-les, d'altres no poden ser alterades pel servidor.

En aquest apartat es farà la llista de la majoria de paràmetres que corresponen a opcions globals de configuració DHCP. Alguns paràmetres són indicats pel client, d'altres pel servidor i n'hi ha que són per defecte i no es poden modificar.

- **option bootfile-name** bootfile-name text; Indica el nom del fitxer que el client s'ha de descarregar per tal d'iniciar el procés d'arrancada o *boot*. El fitxer es descarrega via TFTP del mateix servidor o de l'indicat amb l'opció *next-server*. Fa la mateixa funció que la sentència *filename*.
- **option dhcp-lease-time** dhcp-lease-time uint32; Aquesta opció permet al client demanar al servidor la concessió per a un període concret de temps.
- **option dhcp-message-type** dhcp-message-type uint8; Indica el tipus de missatge DHCP que s'està enviant, tant pel client com pel servidor. Els tipus possibles són:

  - DHCPDISCOVER
  - DHCPOFFER
  - DHCPREQUEST
  - DHCPDECLINE
  - DHCPACK
  - DHCPNAK
  - DHCPRELEASE
  - DHCPINFORM

- **option domain-name** domain-name text; Indica el nom de domini que el client ha d'utilitzar per fer resolucions DNS.
- **option domain-name-servers** domain-name-servers ip-address [, ip-address...]; Especifica la llista de servidors de noms de domini que el client ha d'utilitzar.
- **option host-name** host-name string; Especifica el nom del client.
- **option netbios-name-servers** netbios-name-servers ip-address [, ip-address...]; L'opció NetBios name server (NBNS) especifica una llista de servidors RFC 1001/1002 NBNS name servers en ordre de preferència. És a dir, indica la llista de servidors NetBios a usar pel client.
- **option nis-servers** nis-servers ip-address [, ip-address...]; Especifica una llista de servidors NIS disponibles pel client.
- **option ntp-servers** ntp-servers ip-address [, ip-address...]; Indica una llista de servidors NTP disponibles pel client.
- **option routers** routers ip-address [, ip-address...]; Especifica una llista de routers disponibles pel client en la seva pròpia subxarxa.
- **option tftp-server-name** tftp-server-name text; Indica el nom del servidor TFTP.

1.5.6 Expressions

Una de les grans potències que proporciona DHCP és poder configurar les opcions de xarxa en funció de qui i com és el client. És a dir, en funció de la informació que proporciona el client assignar-li una o altra configuració de xarxa. Fixeu-vos que no es tracta d'entrades *host* estàtiques per a cada client, sinó que un mateix client tindrà una o altra configuració segons la informació que proporcioni.

Avaluar expressions

Per obtenir més informació sobre expressions DHCP i mecanismes d'avaluació cal consultar la pàgina de manual dhcp.eval(5) en un sistema GNU/Linux.

Per poder fer això cal poder avaluar expressions i condicions basades en la informació del mateix client. Existeix, doncs, tot un llenguatge per escriure expressions i avaluar-les.

El mecanisme per definir expressions és:

```
paràmetre = expressió ;
```

Un exemple que mostra com definir el paràmetre ddns-hostname (que defineix el nom del client a usar en l'actualització dinàmica de DNS) usant part de l'adreça MAC del client:

```
ddns-hostname = binary-to-ascii (16, 8, "-", substring (hardware, 1, 6));
```

Es poden construir expressions similars a les dels llenguatges de programació. Per fer-ho es disposa d'estructures condicionals i de diversos tipus d'operadors. Anem a analitzar-ne uns quants:

- **Estructures condicionals** (*Conditional behaviour*). Permeten realitzar definicions segons sigui el valor d'una opció (o d'una expressió). Utilitza les conegudes estructures *if*, *elsif* i *else*.

```
#Exemple de sentència condicional en el servidor
#Extret de la pàgina de manual dhcpd.eval(5)
       if option dhcp-user-class = "accounting" {
         max-lease-time 17600;
         option domain-name "accounting.example.org";
         option domain-name-servers ns1.accounting.example.org,
                         ns2.accounting.example.org;
       } elsif option dhcp-user-class = "sales" {
         max-lease-time 17600;
         option domain-name "sales.example.org";
         option domain-name-servers ns1.sales.example.org,
                         ns2.sales.example.org;
       } elsif option dhcp-user-class = "engineering" {
         max-lease-time 17600;
         option domain-name "engineering.example.org";
         option domain-name-servers ns1.engineering.example.org,
                         ns2.engineering.example.org;
       } else {
         max-lease-time 600;
         option domain-name "misc.example.org";
         option domain-name-servers ns1.misc.example.org,
                         ns2.misc.example.org;
       }
```

En aquest exemple es defineixen les opcions *max-lease-time*, *domain-name* i *domain-name-servers* de manera diferent segons sigui el valor de l'opció *dhcp-user-class* enviat pel client.
Les definicions condicionals també es poden fer en la part client. L'exemple següent mostra que si el client no ha rebut el paràmetre *domain-name*, ell mateix autodefineix el valor del paràmetre *domain-name-servers* al *loopback*.

Exemple de sentència condicional en el servidor

```
#Extret de la pàgina de manual dhcpd.eval(5)
if not option domain-name = "example.org" {
         prepend domain-name-servers 127.0.0.1;
}
```

- **Expressions booleanes** (*Boolean expressions*). Igual que passa en els llenguatges de programació, permeten definir expressions utilitzant els operadors booleans clàssics: =, *and*, *or*, *not*, *exists*. També els operadors *known* i *static*.
- **Expressions de tractament de text** (*Data expressions*). Són similars a les típiques funcions de tractament de cadenes dels llenguatges de programació. Corresponen a: *substring*, *suffix*, *option*, *config-option*, *hardware*, *packet*, *concat*, *reverse*, *leased-address*, *binary-to-ascii*,*encode-int*, *pick-first-value* i *host-decl-name*.
- **Expressions numèriques** (*Numeric expressions*). Les expressions numèriques realitzen la seva avaluació retornant un enter. Són exemples d'aquesta categoria: *extract-int*, *lease-time* i *client-state*.
- **Monitoratge** (*Logging*). Es permeten definir expressions que s'enregistraran en els fitxers de monitoratge del sistema. És molt útil si es treballa amb definicions condicionals i expressions perquè permet fer un seguiment (com un *debug*) de com s'ha avaluat l'expressió o la condició.

```
#Exemple de log
#log (priority, data_expr)
#Les prioritats poden ser: fatal, error, info, debug
log (info, concat(hardware, hostname)
```

Aquest exemple generarà una entrada en el fitxer de *log* mostrant l'adreça Mac del client i el seu nom de *host*.

- **Actualitzacions DNS dinàmiques** (*dynamic DNS updates*). El servei DHCP pot interactuar amb el servei DNS de manera que les concessions DHCP s'actualitzin automàticament en la base de dades DNS. Aquesta funció la poden fer tant els clients com els servidors DHCP. Quan parlem d'actualitzacions dinàmiques del DNS estem parlant de DDNS (*dynamic domain name system*). Com cal fer aquestes actualitzacions i quina informació cal posar al DNS és el que es pot personalitzar amb aquestes opcions.

Per saber més de les actualitzacions dinàmiques de DNS consulteu l'RFC 2136.

1.6 Configuració dels paràmetres de xarxa del client

Els clients de xarxa o bé tenen una configuració estàtica on es defineix cada paràmetre en el client o bé reben la configuració via DHCP. El procés de configurar un client DHCP és tan senzill com activar aquesta última opció usant algun dels mètodes adients.

La configuració dels clients DHCP consisteix en el següent:

- Observar la configuració de xarxa actual del client.

- Configurar el client per rebre dinàmicament una adreça IP. Es tracta d'activar/desactivar la configuració de xarxa dinàmica o estàtica.
- Sol·licitar/renegociar una nova IP al servidor DHCP.
- Observar/fer la llista del fitxer de registre de les concessions client rebudes.
- Activar/desactivar el servei de xarxa en el client.

1.6.1 Observar la configuració de xarxa actual

L'administrador ha de tenir la precaució d'observar quina és la configuració de xarxa actual abans d'establir-ne una de nova. Si cal ha de deixar anotada la configuració anterior per tal de poder restablir-la. La configuració de xarxa bàsica consisteix en l'adreça IP, la màscara de xarxa, la porta d'enllaç per defecte i el servidor DNS a utilitzar. També s'hi pot incloure el nom de l'equip, les rutes i multitud d'altres paràmetres.

Es pot fer la llista de la configuració de les interfícies ethernet i observar les adreces IP, l'estat de les interfícies (UP o DOWN) i les rutes definides fent:

Fer la llista de les adreces MAC i IP i les dades de les interfícies

```
[root@pc]# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:17:31:15:80:7e brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.34/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::217:31ff:fe15:807e/64 scope link
       valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
```

Fer la llista de les rutes definides en el host:

```
[root@pc]# ip route show
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.34
169.254.0.0/16 dev eth0  scope link
default via 192.168.1.1 dev eth0
```

Si s'observen els fitxers de configuració de les interfícies de xarxa, es pot veure si els paràmetres de xarxa estan definits estàticament o dinàmicament. En la llista següent es pot veure que la interfície *loopback* es configura estàticament i la interfície *eth0* es configura via DHCP:

```
[root@portatil ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet

```

```
[root@portatil ~]# cat /etc/sysconfig/network-scripts/ifcfg-lo
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
# If you're having problems with gated making 127.0.0.0/8 a martian,
# you can change this to something else (255.255.255.255, for example)
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

Un error usual és creure que el servei DHCP no funciona correctament quan de fet el que passa és que el client no té el servei de xarxa activat. Es pot comprovar l'estat del servei de xarxa fent:

```
[root@portatil ~]# service NetworkManager status
[root@portatil ~]# service network status
Dispositius configurats:
lo eth0
Dispositius actius actualment:
lo eth0

Activar i desactivar el servei de xarxa:
[root@pc]# /etc/init.d/network stop
S'està  aturant la interficie eth0:                    [ FET  ]
S'està aturant la interficie loopback:                 [ FET  ]

[root@pc]# /etc/init.d/network start
S'està  activant la interficie loopback:               [ FET  ]
S'està  activant la interficie eth0:
S'està  determinant la informació de la IP per a eth0... fet [FET]
```

1.6.2 Configurar el client com a client dinàmic

Tot equip client de xarxa necessita una configuració apropiada. Si aquesta configuració es defineix element per element en el mateix equip, s'anomena configuració estàtica. Si és així, no cal per res un servidor DHCP. És quan els clients reben la configuració de xarxa externament que parlem de configuració dinàmica i ens cal un servidor DHCP que la proporcioni.

La configuració del client es pot fer en mode text editant directament els fitxers de configuració de les interfícies de xarxa, utilitzant interfícies en mode text o utilitzant interfícies gràfiques (*applets*). Vegem cada un d'aquests mètodes.

Editar els fitxers de configuració

Es pot editar directament el fitxer de configuració de la interfície pertinent i establir l'opció *BOOTPROTO* al valor *dhcp* per tal d'activar el client DHCP. Si per exemple es configura la interfície *eth0* seria un fitxer similar al següent:

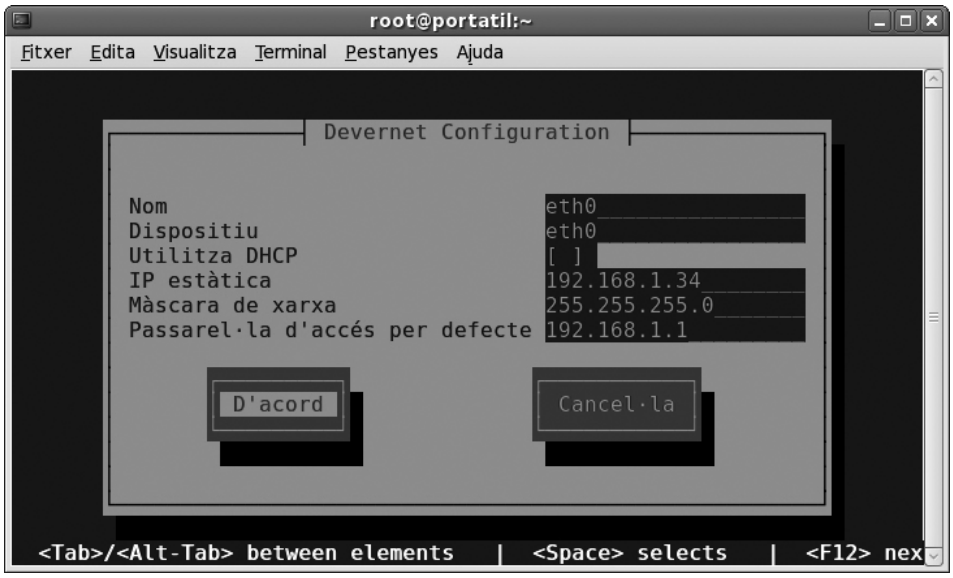
```
[root@portatil ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
```

Menús amb interfície de text

Un altre mecanisme per activar el client DHCP és utilitzar alguna utilitat de menús en entorn de text (varien segons el sistema i n'hi ha de disponibles per Internet).

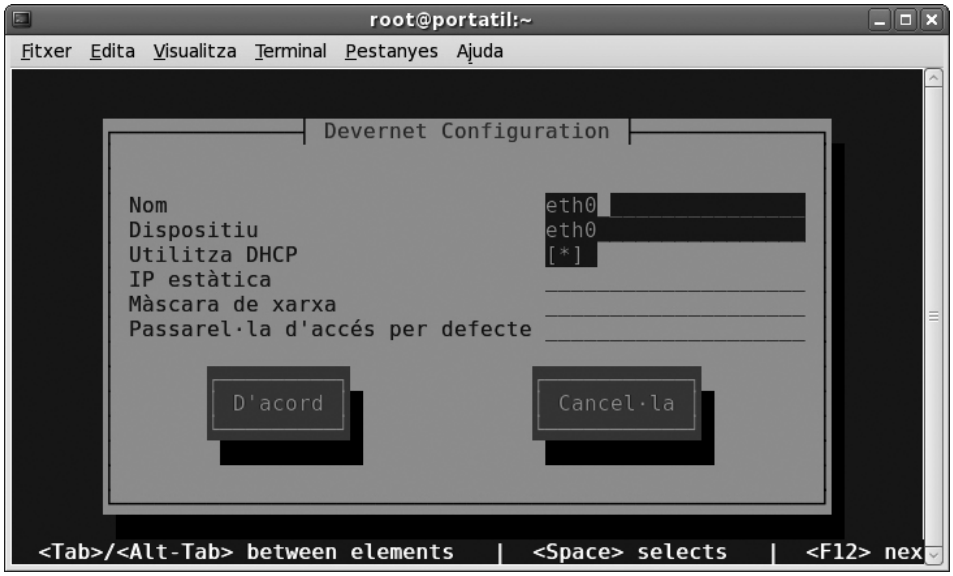
Una eina força utilitzada és la utilitat *setup* que es pot observar en la figura 1.2, que mostra una configuració de client de xarxa estàtica i es pot veure que la casella que permet activar el client DHCP està desactivada.

Figura 1.2. Configuració estàtica del client DHCP



Observeu que senzill que és el procediment per activar el client de xarxa DHCP. N'hi ha prou d'activar l'opció pertinent, tal com es mostra en la figura 1.3.

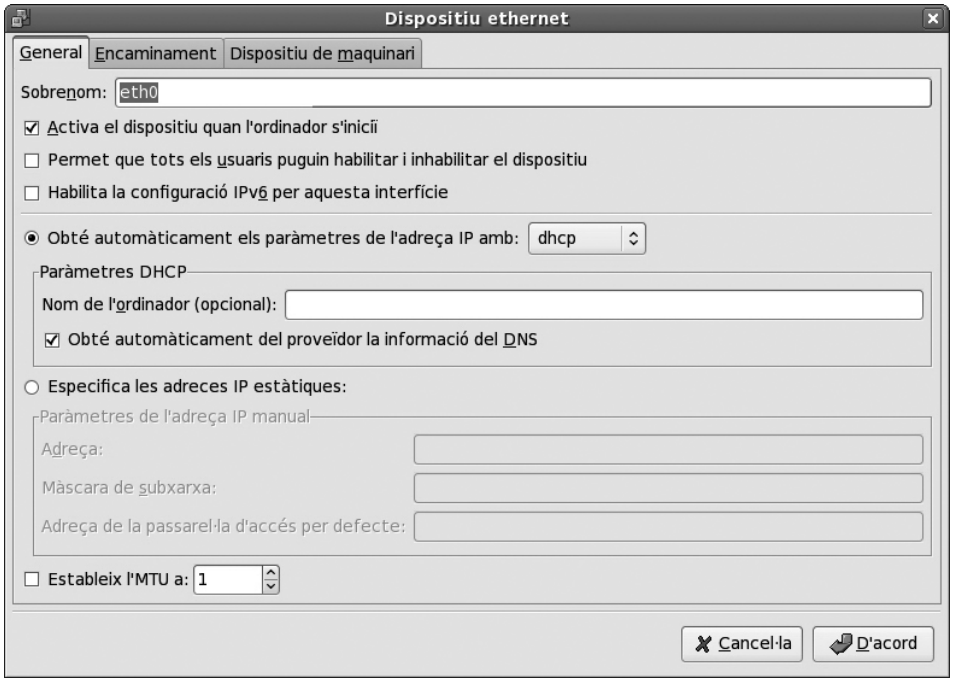
Figura 1.3. Activació del client DHCP usant menús de text



Menús en mode gràfic

En mode gràfic el sistema també proporciona mecanismes per configurar les interfícies de xarxa i establir el mode d'activació a DHCP. En la figura 1.4 es pot observar que la configuració de la interfície *eth0* té activada l'opció per rebre la configuració de xarxa via DHCP.

FIGURA 1.4. Activació del client DHCP usant l'entorn gràfic



1.6.3 Demanar una nova adreça IP

El client DHCP pot alliberar l'adreça que utilitza quan ho creu pertinent. En fer-ho el servidor anota el fi de la concessió i si es tracta d'una adreça dinàmica d'interval aquesta torna a estar disponible per assignar-la a un altre client. Quan a un client se li està acabant el temps de la concessió ha de tornar a negociar una adreça amb el servidor. De totes maneres si el client vol, pot tornar a sol·licitar-ne una en un moment o altre.

El client pot alliberar una adreça (*release*) que està en ús un moment o altre, pot forçar-ho fent, per exemple:

```
[root@portatil ~]# dhclient -r
Internet Systems Consortium DHCP Client V3.0.5-RedHat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Listening on LPF/eth0/00:17:31:15:80:7e
Sending on   LPF/eth0/00:17:31:15:80:7e
DHCPRELEASE on eth0 to 192.168.1.1 port 67
```

Per forçar el client a demanar una nova adreça per a la interfície ethernet *eth0* es pot fer:

```
[root@portatil ~]# dhclient eth0
Internet Systems Consortium DHCP Client V3.0.5-RedHat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Listening on LPF/eth0/00:17:31:15:80:7e
Sending on   LPF/eth0/00:17:31:15:80:7e
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 5
DHCPOFFER from 192.168.1.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.1
bound to 192.168.1.34 -- renewal in 38975 seconds.
```

1.6.4 Observar el registre client de les concessions rebudes

El client DHCP porta un registre de les concessions rebudes, d'aquesta manera pot tornar a demanar una concessió abans que expiri l'actual. També pot servir per demanar al servidor preferentment una adreça IP concreta. Les concessions o *leases* del client es desen en un fitxer anomenat */var/lib/dhclient/dhclient.leases*. Podem veure'n el contingut fent:

```
[root@pc]# cat /var/lib/dhclient/dhclient.leases
lease {
  interface "eth0";
  fixed-address 192.168.1.34;
  option subnet-mask 255.255.255.0;
  option routers 192.168.1.1;
  option dhcp-lease-time 86400;
  option dhcp-message-type 5;
  option domain-name-servers 80.58.61.250,80.58.61.254;
  option dhcp-server-identifier 192.168.1.1;
  option domain-name "local.lan";
  renew 3 2007/12/19 04:05:49;
  rebind 3 2007/12/19 15:12:57;
  expire 3 2007/12/19 18:12:57;
```

1.7 Comprovació del funcionament DHCP

La millor manera de comprovar el funcionament DHCP és simplement posant-lo en pràctica, és a dir, crear una xarxa amb diversos clients DHCP i un servidor que els atengui. Com es pot saber si funciona? Fàcil, mirant un a un cada client i comprovant que han rebut la configuració de xarxa correcta. El problema, però, és què fem si els clients no es configuren correctament.

Els passos més usuals a seguir per a la resolució de problemes són:

- Comprovar que la xarxa està correctament connectada físicament, és a dir cables, connectors, interfícies, etc.

- Mirar si existeix connectivitat entre els equips, per exemple usant una configuració estàtica. Això permetrà descartar que els problemes siguin deguts a altres causes. Si el DHCP no va és que no està configurat correctament.
- Repassar la configuració del client i del servidor DHCP, especialment la del servidor. Es pot començar fent la configuració tan senzilla com sigui possible. Un cop funciona es pot anar avançant en la seva complexitat.
- Examinar els fitxers de concessions, tant el del client com el del servidor, per detectar-hi anomalies.
- Quan la comunicació client/servidor no funciona correctament i no sabem el perquè és molt útil monitorar el trànsit de xarxa mitjançant alguna eina d'anàlisi dels paquets que viatgen per la xarxa.

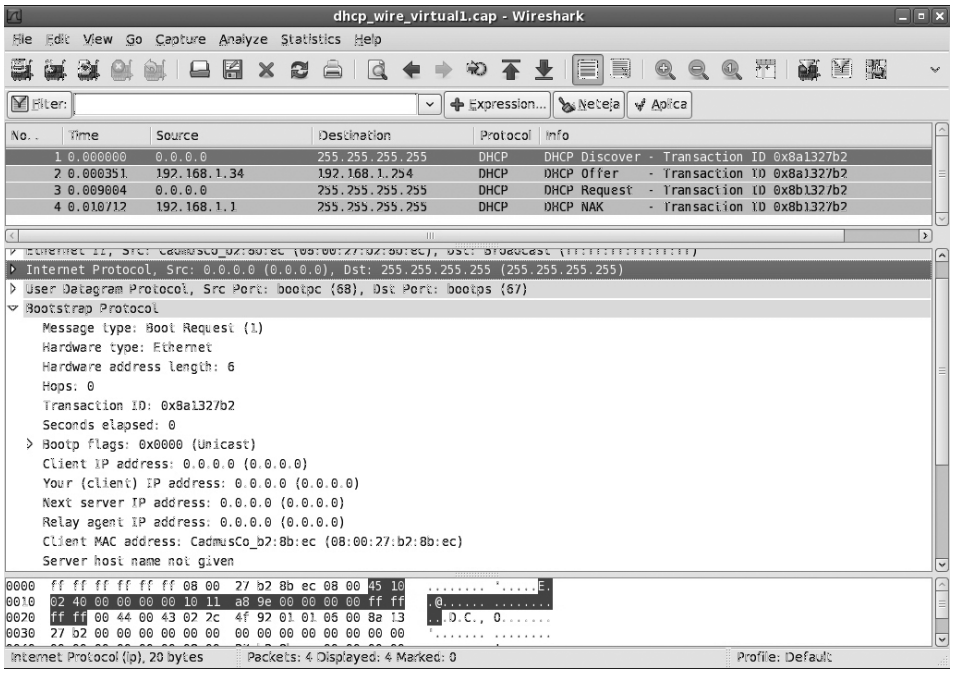
Centrem-nos, doncs, en la monitoratge del trànsit de xarxa per tal d'analitzar que el diàleg entre el client i el servidor és l'apropiat. Existeixen moltes eines al mercat (que podeu trobar per Internet) que fan aquesta funció. Un de les més recomanables és l'aplicació *wireshark*. Amb aquesta eina hem de poder observar l'intercanvi dels paquets *DHCP Discover*, *DHCP Offer*, *DHCP Request* i *DHCP Ack* que es produeix quan tot el procés DHCP funciona correctament. Si aquest intercanvi no es produeix és que existeix algun problema.

En l'apartat "El model funcional del protocol DHCP" podeu observar detalladament com és el diàleg entre el client i el servidor.

En la figura 1.5 podeu observar una captura de trànsit DHCP feta amb *Wireshark*. La captura s'ha fet al servidor i el client s'ha forçat a demanar de nou una configuració de xarxa amb la utilitat *dhclient*.

En l'apartat "Demanar una nova adreça IP" es pot observar com forçar el client a demanar una nova configuració.

FIGURA 1.5. Captura d'un diàleg DHCP client/servidor



Podeu manipular vosaltres mateixos la captura del trànsit de xarxa DNS carregant el fitxer de captura del *wireshark* que es lliura com a material complementari. Aquest fitxer el trobareu en la secció "Annexos" del web del mòdul.

A continuació es pot observar la llista de text de les quatre trames capturades amb el *wireshark* que s'han exportat en format text. Això permet veure detalladament tot el diàleg DHCP.

```
No.     Time        Source                Destination           Protocol Info
      1 0.000000    0.0.0.0               255.255.255.255       DHCP     DHCP
          Discover –
Transaction ID 0x8a1327b2

Frame 1 (590 bytes on wire, 590 bytes captured)
Ethernet II, Src: CadmusCo_b2:8b:ec (08:00:27:b2:8b:ec), Dst: Broadcast (ff:ff:
    ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255
    (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x8a1327b2
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: CadmusCo_b2:8b:ec (08:00:27:b2:8b:ec)
    Server host name not given
    Boot file name not given
    Option: (t=53,l=1) DHCP Message Type = DHCP Discover
    Option: (t=57,l=2) Maximum DHCP Message Size = 548
    Option: (t=55,l=11) Parameter Request List
    Option: (t=12,l=4) Host Name = "box"
    Option: (t=51,l=4) IP Address Lease Time = 12 hours
    End Option
    Padding

No.     Time        Source                Destination           Protocol Info
      2 0.000351    192.168.1.34          192.168.1.254         DHCP     DHCP
          Offer    –
Transaction ID 0x8a1327b2

Frame 2 (342 bytes on wire, 342 bytes captured)
Ethernet II, Src: AsustekC_15:80:7e (00:17:31:15:80:7e), Dst: CadmusCo_b2:8b:ec
(08:00:27:b2:8b:ec)
Internet Protocol, Src: 192.168.1.34 (192.168.1.34), Dst: 192.168.1.254
    (192.168.1.254)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x8a1327b2
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 192.168.1.254 (192.168.1.254)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: CadmusCo_b2:8b:ec (08:00:27:b2:8b:ec)
    Server host name not given
    Boot file name not given
    Option: (t=53,l=1) DHCP Message Type = DHCP Offer
 Option: (t=54,l=4) Server Identifier = 192.168.1.34
    Option: (t=51,l=4) IP Address Lease Time = 12 hours
 Option: (t=1,l=4) Subnet Mask = 255.255.255.0
 Option: (t=3,l=4) Router = 192.168.1.1
    Option: (t=6,l=4) Domain Name Server = 192.168.1.1
    Option: (t=15,l=11) Domain Name = "domain.org"
    End Option
    Padding
```

```
No.     Time        Source                Destination           Protocol Info
      3 0.009004    0.0.0.0               255.255.255.255       DHCP     DHCP
          Request  —
Transaction ID 0x8b1327b2

Frame 3 (590 bytes on wire, 590 bytes captured)
Ethernet II, Src: CadmusCo_b2:8b:ec (08:00:27:b2:8b:ec), Dst: Broadcast (ff:ff:
    ff:ff:ff:ff)
Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255
    (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
    Message type: Boot Request (1)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x8b1327b2
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: CadmusCo_b2:8b:ec (08:00:27:b2:8b:ec)
    Server host name not given
    Boot file name not given
    Option: (t=53,l=1) DHCP Message Type = DHCP Request
    Option: (t=57,l=2) Maximum DHCP Message Size = 548
    Option: (t=55,l=11) Parameter Request List
    Option: (t=12,l=4) Host Name = "box"
    Option: (t=51,l=4) IP Address Lease Time = 12 hours
    Option: (t=54,l=4) Server Identifier = 192.168.1.34
    Option: (t=50,l=4) Requested IP Address = 192.168.1.254
    End Option
    Padding

No.     Time        Source                Destination           Protocol Info
      4 0.010712    192.168.1.1           255.255.255.255       DHCP     DHCP
          NAK      —
Transaction ID 0x8b1327b2

Frame 4 (342 bytes on wire, 342 bytes captured)
Ethernet II, Src: XaviTech_7b:ff:1d (00:01:38:7b:ff:1d), Dst: Broadcast (ff:ff:
    ff:ff:ff:ff)
Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 255.255.255.255
    (255.255.255.255)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
    Message type: Boot Reply (2)
    Hardware type: Ethernet
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x8b1327b2
    Seconds elapsed: 0
    Bootp flags: 0x8000 (Broadcast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 192.168.1.1 (192.168.1.1)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: CadmusCo_b2:8b:ec (08:00:27:b2:8b:ec)
    Server host name not given
    Boot file name not given
    Option: (t=53,l=1) DHCP Message Type = DHCP NAK
    Option: (t=56,l=31) Message = "requested address not available"
    End Option
    Padding
```

1.8 Realitzar documentació de suport a l'usuari

Una de les facetes més ignorades en el camp de la informàtica és la confecció de manuals i documentació de suport. Com a clients molt sovint ens queixem que falta informació o que està mal redactada. Com a administradors de xarxa no trobem mai temps per anotar les coses. Mentre les tenim al cap no creiem necessari fer la documentació, quan no ho tenim al cap, ja ens és impossible fer-ho, i sovint, és just quan en faria falta haver-ho fet! Fem un repàs de la informació necessària que cal documentar, tant per a l'usuari com per a l'administrador.

El client ha de saber:

- Com contactar amb el servidor DHCP. Quin programari ha d'utilitzar i com l'ha de configurar per fer ús del servei.
- Quina és la informació que s'obtindrà via DHCP. Cal saber visualitzar, consultar aquesta informació i saber què significa, per a què serveix.

Un altre exemple de captura DHCP es lliura com a material complementari. En aquesta captura hi ha dos servidors dhcp diferents en la xarxa i es poden observar paquets NACK. Aquest fitxer el trobareu en la secció "Annexos" del web del mòdul.

Així doncs, la documentació de l'usuari descriurà el procés per activar el client DHCP, amb captures de pantalla que facilitin aquest procés. Cal també exemples de llista de concessions rebudes, on són i com es poden consultar aquestes concessions. La part més important és mostrar un exemple de configuració de xarxa rebuda on es detalli el significat de cada element i explicar a l'usuari com fer aquesta consulta.

Un exemple d'informació a proporcionar podria ser el següent:

```
L'usuari pot consultar la concessió de xarxa amb l'ordre:
C:\>ifconfig /all
Configuración IP de Windows
        Nombre del host . . . . . . . . . . : nombre-29b9943f
        Sufijo DNS principal  . . . . . . . :
        Tipo de nodo. . . . . . . . . . . . : híbrido
        Enrutamiento habilitado. . . . . .: No
        Proxy WINS habilitado. . . . .    : No
        Lista de búsqueda de sufijo DNS:    local.lan

Adaptador Ethernet Conexiones de red inalámbricas          :
        Estado de los medios. . . .: medios desconectados
        Descripción. . . . . . . . . . . . : Intel(R) PRO/Wireless 3945ABG Net
            Connection
        Dirección física. . . . . . . . . : 00-31-02-44-9F-5A

Adaptador Ethernet Conexión de área local          :
        Sufijo de conexión específica DNS : local.lan
        Descripción. . . . . . . . . . . . : Realtek RTL8168/8111 PCI-E Gigabit
            Ether NIC
        Dirección física. . . . . . . . . : 00-21-32-80-23-7D
        DHCP habilitado. . . . . . . . . . : No
        Autoconfiguración habilitada. . . : Sí
        Dirección IP. . . . . . . . . . . . : 192.168.1.33
        Máscara de subred . . . . . . . . . : 255.255.255.0
        Puerta de enlace predeterminada   : 192.168.1.1
        Servidor DHCP . . . . . . . . . . . : 192.168.1.1
        Servidores DNS . . . . . . . . . . .: 80.58.61.250
                                              80.58.61.254
```

```
28          Concesión obtenida . . . . . .  : jueves, 24 de septiembre de 2009
                12:09:16
29          Concesión expira . . . . . .  . .: jueves, 24 de septiembre de 2009
                13:09:16
```

Els principals valors de xarxa a destacar són: Adreça IP, màscara de xarxa, el router o porta d'enllaç predeterminada, el servidor dhcp amb el qual s'ha contactat, els servidors de noms DNS que s'utilitzen i quan expira la concessió.

2. Instal·lació de serveis de resolució de noms

El sistema de noms de domini **DNS** (*domain name system*) proporciona un mecanisme eficaç per fer la resolució de noms de domini a adreces IP. Com a usuaris (humans) ens és més fàcil adreçar-nos a un nom de domini (de *host*, de web, de servidor de correu, etc.) utilitzant un text identificatiu (per exemple, www.ioc.cat) que no pas a l'adreça IP pertinent (per exemple, *213.73.40.230*). El servei DNS no solament permet fer la resolució de noms de domini a adreces IP, sinó també la resolució inversa. És a dir, a partir d'una IP esbrinar el nom de domini.

El servei DNS proporciona independència del nom de domini respecte a la IP. Així un domini pot canviar d'IP de manera transparent per als usuaris del domini. Fins i tot és usual que un domini s'identifiqui amb més d'una IP com a mesura de redundància contra la caiguda del sistema o com a balanceig de càrregues. Altres serveis proporcionats pel DNS són la identificació dels servidors de correu d'un domini, de cada un dels *hosts* que pertanyen a la xarxa, servidors d'impressió, etc.

2.1 Sistemes de noms plans i jeràrquics

El problema d'identificar els equips es produeix des de bon principi de l'existència de les xarxes d'ordinadors i no és específic de les xarxes TCP/IP. Cal un mecanisme en "llenguatge humà" per identificar els equips de la xarxa. En especial els que proporcionen serveis als altres equips i usuaris. En la xarxa inicial ARPANET, els equips ja rebien un nom. Aquests noms es feien públics per mitjà d'un fitxer centralitzat que contenia els noms de tots els equips de la xarxa i la seva identificació. Aquest fitxer era el fitxer *hosts.txt* (en sistemes GNU/Linux, conegut com a /etc/hosts).

ARPANET

ARPANET és la xarxa de commutació de paquets que va desenvolupar el Departament de Defensa dels Estats Units i que va ser la predecessora d'Internet.

Un sistema de **noms pla** es basa en la utilització d'un **fitxer de text** que descriu cada *host* amb la seva corresponent adreça IP. Es pot usar per definir àlies per equips locals en xarxes petites, però no és escalable a xarxes grans i molt menys a Internet.

Exemple de fitxer de noms pla per descriure els àlies d'una xarxa local a la llar:

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

Posar noms als ordinadors de casa:

El fitxer anterior pertany al PC anomenat *server1* i posa noms "casolans" als altres ordinadors de la família. Així per exemple l'equip *192.168.1.33* es pot anomenar de les quatre maneres diferents que s'indiquen.

En una xarxa petita es pot generar un fitxer amb el nom i identificador IP de tots els *hosts*, centralitzat en un servidor, i encarregar-se de distribuir còpies d'aquest fitxer a tots els equips de la xarxa. Però aquest model de coneixement no és escalable. Si la xarxa creix és impossible de mantenir. Utilitzar aquest model significaria que hi ha un equip que centralitza els noms de tots els *hosts* d'Internet en un sol fitxer! D'altra banda, també significaria que aquest fitxer s'ha de repartir entre tots els equips d'Internet perquè sàpiguen com es diuen els altres equips cada cop que hi ha una actualització. Evidentment cal una altra solució.

Problemes d'actualització d'una xarxa

Imagineu els problemes que es presenten en voler fer arribar un mateix fitxer a tots els *hosts*. Hi haurà inconsistències amb equips que han rebut l'actualització i altres que no. El trànsit de copiar el fitxer en cada equip. Un nou equip a Internet vol dir escriure de nou el fitxer i tornar-lo enviar a tothom!

El 1983 sorgeix el *domain name system* (DNS) per aportar una solució escalable i pràctica. El DNS es basa en una base de dades de noms de domini jeràrquica i distribuïda. **Jeràrquica** perquè s'organitza en una estructura de **dominis** que es poden compondre de subdominis que també es poden dividir en subdominis i així fins a 127 nivells (originàriament). Aquests dominis són gestionats per servidors DNS responsables de cada **zona**. És una base de dades **distribuïda** perquè la informació no està tota junta en un sol repositori central, sinó que la informació es troba repartida per parts en els servidors DNS d'Internet. Cada servidor DNS **autoritari** conté la base de dades de la seva zona.

2.1.1 Elements del sistema de noms de domini

El servei DNS permet identificar qualsevol equip en la xarxa i assegurarse que no hi ha col·lisions, és a dir, noms duplicats. Es basa en una estructura jeràrquica de noms en forma d'arbre on l'arrel és el node o domini arrel del qual deriven tots els altres nodes. Aquest es divideix en altres dominis com, per exemple, *.com*, *.edu*, *.org*, *.cat*, etc. Al seu torn, cada domini es pot dividir en altres subdominis i així successivament. Les rutes s'indiquen començant pel subdomini més intern cap al node arrel (mail.ioc.cat).

Caràcters en els noms de domini

L'estàndard DNS indica que els noms de domini han de ser de seixanta-quatre caràcters com a màxim, i només poden incloure caràcters llatins, dígits del 0 al 9 i el guió. Les majúscules i minúscules són indiferents. Hi ha mecanismes com l'IDNA (*an internationalized domain name* o noms de domini internacionalitzats) per permetre utilitzar altres alfabets en els noms de domini.

Els nodes s'identifiquen per un text (el nom de domini) que no es pot repetir en el mateix nivell, però sí en altres llocs de l'arbre de l'espai de noms. El mateix passa amb els fitxers: no hi pot haver dos fitxers amb el mateix nom en el mateix directori, però sí en ubicacions diferents. Un **domini** és el node indicat i tota la resta de l'arbre que penja d'aquest node (penseu en l'exemple d'un directori: si es vol copiar un directori, s'entén que està format pel mateix directori i tots els subdirectoris que conté). S'entén per **espai de noms** el conjunt de tots els dominis que formen l'arbre DNS.

El sistema de noms de domini d'Internet DNS utilitza els elements següents:

- **Espai de noms**. El conjunt de tots els dominis (l'arbre).
- **Domini**. Text identificatiu d'un domini.
- **FQDN**. Nom de domini absolut començant pel node i acabant en l'arrel.
- **Domini absolut**. (FQDN) Els dominis absoluts acaben en punt (.).
- **Domini relatiu**. Nom de domini sense qualificar.
- **Domini arrel**. Domini del qual deriven tots els altres. S'indica amb un punt o amb la cadena buida.

Arbres dels sistemes DNS i de fitxers

El sistema de noms de domini té una estructura similar a l'arbre que s'utilitza per representar un sistema de fitxers. La diferència rau en el fet que les rutes dels sistemes de fitxers s'indiquen des de l'arrel fins al node (per exemple *./etc/bind/bind.conf*), i en canvi en el DNS les rutes s'indiquen del node inferior a l'arrel (ns1.ioc.cat).

Dominis d'alt nivell

Els següents són exemples de dominis d'alt nivell d'alguns països:

- CAT: Catalunya.
- AD: Andorra.
- AQ: Antàrtida.
- GB: Gran Bretanya.
- UK: Regne Unit.
- IM: Illa de Man.
- MS: Montserrat.
- PF: Polinèsia Francesa.
- PS: autoritat palestina.

L'estructura en arbre (jeràrquica) de l'espai de noms proporciona un mecanisme d'identificació únic d'un domini. No pot existir cap domini que tingui exactament el mateix nom absolut **FQDN** (full qualified domain name o **nom de domini complet**). Els dominis es llegeixen des del node a l'arrel. Així un domini que correspongui al departament d'administració de l'organització IOC dins del domini *cat* s'identifica, per exemple, com a admin.ioc.cat. Si ens fixem en el domini anterior, veurem que acaba en punt, és una manera d'indicar el domini arrel. El **domini arrel** es defineix com un domini sense etiqueta, o millor amb la cadena buida com a etiqueta. Això provoca que els dominis que s'indiquin de manera absoluta acabin amb el caràcter punt.

Un **domini absolut** o FQDN és el que inclou tots els nodes des del domini fins a l'arrel (inclosa en forma de punt final). Un **domini relatiu** no inclou l'arrel i pot ser relatiu al domini actual. Per exemple, dins del domini de

l'IOC el domini *inf* (del departament d'informàtica) és un nom relatiu que fa referència al nom absolut inf.ioc.cat.

El punt final en els noms de domini

La majoria de vegades escrivim els dominis com si fossin absoluts, però són relatius al node arrel perquè no posem el punt final. Un altre cop es pot fer l'analogia amb les rutes relatives i les rutes absolutes del sistema de fitxers.

2.1.2 Els noms de domini d'Internet

A Internet els noms de dominis segueixen una estructura basada en els seus inicis, però que ha anat evolucionant. El node arrel es va dividir en un conjunt de subdominis anomenats **TLD** (top level domains o **dominis d'alt nivell**). Aquests dominis eren *com*, *edu*, *gov*, *mil*, *org*, *net* i *int*. Posteriorment se'n van afegir d'altres com *cat*, *name*, *biz*, *info*, *pro*, *aero*, *coop* i *museum*. Es volien organitzar els dominis per funcionalitat posant les empreses en els *.com*, les organitzacions en els *.org*, etc. Es va veure, però, la necessitat de poder agrupar els dominis de manera geogràfica i van sorgit els famosos identificadors de país. Per a cada país es va generar un TLD de dos caràcters utilitzant el ja existent estàndard internacional ISO 3166 (els famosos *.es*, *.fr*, etc.).

Origen dels noms de domini

Si ens fixem en els primers dominis d'alt nivell, estaven basats en una visió estatunidenca del món (de fet la xarxa ARPANET base de l'actual Internet era seva). En estendre Internet globalment i aparèixer dominis d'alt nivell geogràfics, moltes organitzacions es van registrar en més d'un domini (per exemple, *empresa.com* i *empresa.cat*).

Els servidors arrel són crucials per al funcionament del DNS, ja que coneixen tots els dominis de primer nivell. Han d'admetre un gran volum de consultes i per això n'hi ha tretze repartits per tot el món. A més a més, d'aquests tretze, alguns tenen rèpliques en diversos continents utilitzant un sistema anomenat *anycast*.

2.2 Zones primàries/secundàries i mecanisme de resolució

Exemple d'administració de subdomini

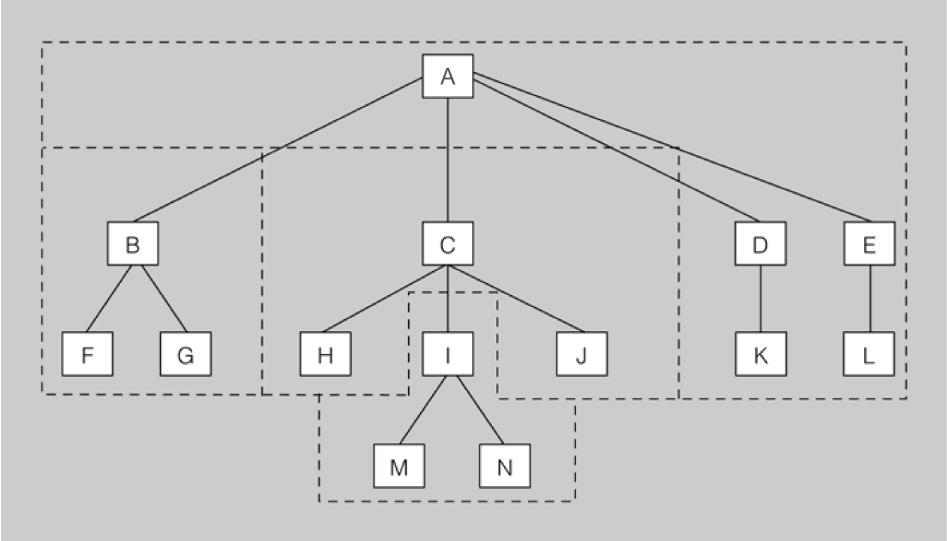
El domini *.cat* és administrat per una entitat que gestiona la zona *.cat*. Aquest domini conté el subdomini *ioc.cat*, però ha **delegat** l'administració d'aquest subdomini a l'IOC. Els administradors de l'IOC disposen d'un servidor que gestiona el seu domini com una zona. El domini *.cat* és l'arbre que inclou tots els dominis que en deriven, també el ioc.cat. Però la zona *.cat* i la zona *ioc.cat* no són la mateixa zona. Són administrades per entitats diferents.

De bon començament sabem que el sistema de noms de domini està basat en una arquitectura client-servidor en què els clients faran preguntes del tipus "quina IP té aquest domini?", i els servidors miraran de contestarles. Els servidors de noms DNS són els programes que emmagatzemen i gestionen la informació en la base de dades d'una part de l'espai de noms anomenada *zona*.

2.2.1 Les zones

Primerament hem de descriure què és una zona. Una **zona** és part de l'espai de noms de domini gestionada per un (o més) servidors DNS. Els servidors que gestionen la zona tenen informació completa sobre la zona i es diu que tenen **autoritat** respecte a ella. De bon principi podríem pensar que un servidor DNS gestiona un domini i que una zona és el mateix que un domini, però això no ha de ser necessàriament així. Un domini es divideix en subdominis per facilitar-ne l'administració, i cada part administrada per un (o més) servidor DNS és una zona. El domini és l'arbre de l'espai de noms i la zona és la part de l'arbre administrada per un servidor de noms de domini concret. En la figura 2.1 es pot veure un espai de noms amb quatre zones i catorze dominis.

FIGURA 2.1. Exemple de zones i dominis



En la figura es poden veure tants dominis com requadres de lletres agrupats en quatre zones. El nom de domini corresponent a cada zona (s'anomenen segons el seu node superior) és A, B.A, C.A, I.C.A. Cada una d'aquestes quatre zones tindrà un (o més) servidors DNS per gestionar-la.

En general, podem dir que una zona conté la informació completa dels equips que formen el domini corresponent a la zona i dels equips dels subdominis que no s'hagin delegat. Aquesta informació s'emmagatzema en la **base de dades de zona**.

Convé tenir clar en tot moment que domini i zona no són equivalents (tot i que poden coincidir).

- El **domini** és l'arbre de l'espai de noms.
- La **zona** és la part de l'arbre administrada per un servidor DNS concret.
- La **base de dades de zona** la formen els fitxers que emmagatzemen la descripció dels equips que pertanyen a la zona.
- La **delegació** consisteix a passar l'autoritat de la gestió d'un subdomini a una altra entitat.

Delegar l'administració d'un subdomini no és més que passar l'autoritat sobre aquest subdomini a una altra entitat (a uns altres servidors DNS). Aquesta entitat és la responsable de l'administració de la zona delegada. Té tota l'autoritat per fer i desfer al seu criteri. La zona pare perd el control administratiu de la zona delegada i simplement apunta als servidors de noms de la zona delegada per obtenir informació quan la requereix.

L'estàndard que defineix el DNS estableix que cal configurar dos o més servidors autoritaris per a cada zona anomenats *servidor primari* i *servidor secundari*. El motiu és proporcionar un mecanisme de redundància, robustesa, rendiment i còpia de seguretat. Si el servidor de noms falla i és únic possiblement la xarxa caurà, serà inoperativa.

Podeu trobar més informació sobre els mecanismes de transferència entre servidors primaris i secundaris, i sobre el concepte de servidor autoritari, en el subapartat "Realització de transferències entre dos o més servidors" d'aquesta unitat.

Els servidors **primari** i **secundari** són autoritat. Només el primari té els fitxers de zona. El secundari n'obté una còpia per transferència.

2.2.2 La resolució de noms

Tot sovint en les aplicacions d'usuari i de sistema s'accedeix a recursos pel seu nom de domini. Per exemple, un client web requereix una determinada pàgina web, un navegador de fitxers vol accedir a unes carpetes d'una màquina remota que s'identifiquen pel nom de domini, el sistema ha de validar l'usuari contra un servidor LDAP remot, etc. En cada un d'aquests casos caldrà resoldre una pregunta del tipus *aquest domini a quina adreça IP correspon?* Aquesta pregunta no la responen les aplicacions individualment (el navegador web, el client d'autenticació...), sinó que utilitzen el resolver per fer-ho.

El *resolver* és la part client de l'arquitectura client-servidor del DNS. Ell ha d'atendre les necessitats de les aplicacions, confeccionar una consulta o *query*, fer-la a un servidor DNS, obtenir la resposta i passar-la a l'aplicació pertinent. El *resolver* no és usualment una aplicació sinó un conjunt de biblioteques de funcions. Les aplicacions clients es compilen i enllacen conjuntament amb aquestes biblioteques.

La resolució

El mecanisme de resolució de noms DNS consta d'un client o *resolver* que realitzarà les consultes (o *querys*) a resoldre a uns servidors DNS.

Si el servidor disposa de la informació perquè forma part de la base de dades de la seva zona, emetrà una resposta **autoritativa**. Si disposa de la resposta perquè la té emmagatzemada temporalment (en un procés anomenat *cache*) també emetrà la resposta però aquest cop de manera**no autoritativa**. Si no té informació del domini buscat, el servidor pot fer a altres servidors la mateixa consulta en un procés que pot ser **recursiu** o **iteratiu**. Sempre existeix un camí per trobar el domini buscat, que és preguntar als **nodes arrel** (*root servers*) de l'espai de noms de domini. Partint dels nodes arrel i recorrent l'arbre cap avall, es pot arribar al domini buscat, si és que existeix.

Sempre hi ha un camí a un domini existent partint del node arrel. Quan un servidor és consultat sobre un domini que desconeix (no és de la seva zona ni té la resposta en la *cache*) pot escalar la pregunta a un servidor de l'arrel (*root name server*). Això significa que els servidors arrel són crucials per al funcionament del DNS.

Exemple de resolució de noms DNS

Quina adreça IP té el domini *ns1.ioc.cat*?

Si un estudiant australià intenta esbrinar això des del seu servidor de noms de Sindney, probablement acabarà preguntant a un dels nodes arrel per aquest domini. El node arrel desconeix el *host ns1* del domini de l'IOC, però sí que coneix tots els dominis de primer nivell (TLD). Proporcionarà una llista amb els servidors de noms del domini *.cat*. Preguntant a algun d'aquests servidors (del domini *.cat*) s'obtindrà la llista de servidors DNS del domini*ioc.cat*. Preguntant als servidors d'aquest domini s'obtindrà l'adreça IP del *host ns1* pel qual el domini *ioc.cat* és autoritari (forma part de la seva zona).

Recursió i iteració

Quan el client o *resolver* emet una consulta al servidor DNS local (el servidor de noms que té configurat), aquest la pot tractar de manera *recursiva* o *iterativa*. De fet, el client *resolver* ja farà la consulta indicant si exigeix una resposta recursiva o iterativa. La diferència entre un mode i l'altre és com ha d'actuar el servidor DNS per obtenir la resposta quan no la té en la seva base de dades d'informació.

En mode **iteratiu** el servidor retorna la millor resposta possible basada en la seva informació local, sense preguntar a ningú més. En el mode **recursiu** el servidor intenta trobar la resposta preguntant a tants altres servidors com calgui per tal d'obtenir-la.

Un servidor pot emetre les respostes següents:

1. Respon enviant la dada que li han sol·licitat (un nom de *host*, una adreça IP, la llista de servidors de noms, de servidors d'autenticació...).
2. Ha localitzat el domini buscat, però no es disposa de la dada sol·licitada. Cal tenir en compte que es poden sol·licitar altres dades a part de l'adreça IP del domini (per exemple, quins servidors de correu té el domini).
3. Finalment pot ser que el domini sol·licitat no existeixi.

Recursió

Quan un servidor rep una consulta del client mira la base de dades local de la seva zona. Si existeix la informació sol·licitada la retorna. Si la dada no forma part de la seva zona, però la té emmagatzemada en *cache* (per què ja ha realitzat amb anterioritat un consulta similar i ha emmagatzemat temporalment la resposta) també la retorna. Si la dada no forma part del seu espai de noms ni es troba en la *cache*, el mode recursiu mana al servidor anar preguntant recursivament a altres servidors, apropant-se més a cada pas al domini sol·licitat. Si el servidor no coneix cap servidor més proper al domini buscat a qui preguntar, acaba preguntant als servidors de l'arrel.

Exemple de recursió

Si es consulta el domini www.inf.ioc.cat i el servidor desconeix aquest domini, intentarà contactar amb un servidor de noms del domini *inf.ioc.cat*. Si tampoc sap com adreçar-se a aquest domini, intentarà contactar amb un servidor de noms de domini *ioc.cat*. Un altre cop si el desconeix provarà de localitzar un servidor per al domini *.cat*. Si tampoc és el cas, es posarà en contacte amb un servidor de noms de l'arrel. Un cop en l'arrel, sempre és possible accedir al domini buscat descendint per l'arbre de dominis.

Si tots els processos recursius acabessin preguntant als nodes arrel, aquests es saturarien. El servidor que ha rebut la consulta del *resolver* pregunta al node més proper al domini buscat. Si coneix algun servidor de noms més proper, li ho pregunta i s'evita d'anar a l'arrel.

Una altra manera d'evitar la sobrecàrrega dels nodes arrel és l'ús de la informació emmagatzemada de consultes anteriors, que es desa localment en la *cache* del servidor.

Consulta d'informació delegada

Si es consulta l'adreça www.inf.ioc.cat i el servidor que rep la consulta és el servidor de noms del domini*ioc.cat*, aquest no pregunta cap amunt (a *.cat* o a l'arrel) sinó que obté de la seva pròpia base de dades la llista dels servidors de noms autoritaris de la zona delegada*inf.ioc.cat*, als quals preguntarà per obtenir una resposta.

Imagineu un alumne de Sydney estudiant de l'IOC que genera una consulta al servidor de noms del seu ISP australià per identificar el domini www.int.ioc.cat. Probablement el servidor desconeix aquest domini i tots els més propers, *inf.ioc.cat* i *ioc.cat*. Però segurament en la *cache* (per altres consultes) té la llista de servidors de noms autoritaris del domini *.cat*. Serà a un d'aquests servidors (i no un node arrel) a qui farà la consulta que li permetrà accedir descendentment al domini buscat.

Per tant, en el procés recursiu, el servidor de noms que rep la consulta del*resolver* ha de tornar una resposta que pot procedir de la seva base de dades de zona, de la *cache*, o de les respostes finals que ha obtingut preguntant recursivament a altres servidors més propers al domini a consultar.

Els nodes arrel no accepten consultes recursives per evitar-ne l'abús i la saturació.

Fixeu-vos que un servidor que rep una consulta recursiva del *resolver* té la feina d'esbrinar per a ell la resposta. Podria repetir la mateixa consulta al servidor més proper fent-la recursiva en lloc d'iterativa. Això exigiria a l'altre servidor fer tota la feina. Aquest plantejament, tot i que possible, es considera abusiu.

Usualment el **client** consulta el seu DNS de manera **recursiva**, i els **servidors** es consulten entre ells de manera **iterativa**.

Iteració

En mode iteratiu, un servidor dóna la millor resposta possible basant-se en la pròpia informació (base de dades de zones locals i del *cache*). En cap

cas consulta cap altre servidor. Si no disposa de la resposta, lliura una llista amb els servidors més propers al domini que es busca. La llista pot ser d'un o més servidors i és tasca del servidor que ha fet la consulta decidir a quin d'ells tornar a preguntar (en el cas recursiu).

Les consultes iteratives són usualment de servidor a servidor, però no del *resolver* al servidor. Si el *resolver* fes una consulta iterativa a un servidor, significaria que

quan la resposta fos una referència a un altre servidor, el *resolver* hauria de fer una altra consulta. Generalment els *resolver* no tenen aquesta capacitat, simplement fan una consulta recursiva al servidor que tenen configurat i és aquest el que ha de fer tota la feina per obtenir la resposta.

El client *resolver* fa una consulta **recursiva** al seu servidor DNS local. Si el servidor DNS disposa de la resposta, la torna . Pot ser de la seva zona i serà una resposta **autoritativa** o pot tenir-la en*cache* i serà **no autoritativa**.
Si no disposa de la resposta, consulta **iterativament** altres servidors apropant-se al domini buscat. Cada servidor que consulta iterativament li pot proporcionar la resposta (autoritativa o no) si la coneix, o una llista de servidors DNS autoritatius per al domini indicat.

Resolució inversa

El DNS proporciona un mecanisme per obtenir el nom de domini a què correspon una adreça IP. Aquest mecanisme, anomenat **resolució inversa**, es basa en un domini especial anomenat *IN-ADDR.ARPA*. Hi ha protocols de xarxa que requereixen una resolució inversa correcta per funcionar bé i sovint s'utilitza com a mesura de seguretat per verificar l'existència de l'adreça IP en un domini.

Un host amb l'adreça IP 192.168.1.24 correspon al domini 24.1.168.192.IN-ADDR.ARPA.

S'ha ideat un domini de nom *IN-ADDR.ARPA* que permet representar en forma de nom de domini totes les adreces IP possibles. El format són etiquetes numèriques del 0-255 que representen cada octet d'una adreça IP. Les etiquetes dels octets es concatenen en ordre invers i se'ls afegeix el sufix *IN-ADDR.ARPA*. Un nom de domini amb quatre etiquetes d'octets correspon a un*host*, un nom de domini amb menys etiquetes correspon a una xarxa.

La xarxa 172.16.32.0/24 correspon al domini ARPA següent: 32.16.172.IN-ADDR.ARPA.

En l'exemple següent es mostren els servidors de noms del domini ioc.cat i es fa una consulta de resolució inversa a cada un d'ells:

```
[root@localhost ~]# host -vt NS ioc.cat
Trying "ioc.cat"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6116
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;ioc.cat.                       IN      NS

;; ANSWER SECTION:
ioc.cat.                900     IN      NS      dns1.nominalia.com.
ioc.cat.                900     IN      NS      dns2.nominalia.com.

;; ADDITIONAL SECTION:
dns1.nominalia.com.     145929  IN      A       62.97.103.68
dns2.nominalia.com.     145925  IN      A       213.151.118.169

Received 108 bytes from 80.58.61.250#53 in 112 ms

[root@localhost ~]# host 62.97.103.68
68.103.97.62.in-addr.arpa domain name pointer dns1.nominalia.com.

[root@localhost ~]# host  213.151.118.169
169.118.151.213.in-addr.arpa domain name pointer dns2.nominalia.com.
```

2.2.3 El protocol DNS

El servei de noms de domini utilitza el protocol DNS per fer les consultes i les respostes. Es tracta d'un protocol de capa d'aplicació que pot utilitzar tant UDP com TCP en la capa de transport. Usualment, tant les consultes del client com les respostes del servidor es poden encabir en un datagrama (512 bytes) i s'utilitza UDP (de fet, generalment es diu que el DNS usa UDP). Però si la informació a transmetre és àmplia (per exemple, una resposta amb una llista amb molta informació), la comunicació es passa a TCP automàticament. Un altre cas en què la informació és TCP és quan es realitza la transferència d'informació d'una zona entre servidors primaris i secundaris. El servidor DNS utilitza el port privilegiat 53.

> El **protocol DNS** és usualment UDP, però pot ser **TCP i UDP**. Es tracta d'un protocol de capa d'aplicació i utilitza el **port 53**.

Els datagrames DNS es componen de diversos apartats tal com es pot veure a la consulta host següent:

```
[root@portatil ~]# host -a uoc.es
Trying "uoc.es"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14091
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;uoc.es.        IN  ANY

;; ANSWER SECTION:
uoc.es.     82747 IN  NS  tibet.uoc.es.
uoc.es.     82747 IN  NS  nepal.uoc.es.

;; AUTHORITY SECTION:
uoc.es.     82747 IN  NS  nepal.uoc.es.
uoc.es.     82747 IN  NS  tibet.uoc.es.

;; ADDITIONAL SECTION:
nepal.uoc.es.   73649 IN  A 213.73.40.47
tibet.uoc.es.   76582 IN  A 213.73.40.45

Received 124 bytes from 127.0.0.1#53 in 2 ms
```

La comunicació DNS és un mecanisme de consulta/resposta entre el client i el servidor. Els datagrames, doncs, seran de *query* (consulta) o *answer* (resposta).

Els apartats que componen un missatge DNS són:

- **HEADER**. Capçalera del missatge indicant si és una consulta o una resposta. Conté l'*id* (identificador) del missatge, *flags* i un resum de quines seccions del missatge porten informació i quanta.
- **QUESTION**. Aquesta secció conté la consulta que s'ha efectuat. És a dir, quina dada s'ha demanat al servidor. Pot ser una resolució d'adreça IP a un domini, demanar la llista de servidors d'impressió, etc.

- **ANSWER**. Secció que conté la resposta obtinguda del servidor. S'entén que aquesta secció conté la resposta no autoritativa. A vegades en les utilitats de consulta aquesta secció es mostra com a *non-autority answer*.
- **AUTHORITY**. Aquesta secció conté les respostes que són autoritatives per a la consulta efectuada. Evidentment pot ser buida.
- **ADITIONAL**. Conté informació addicional per completar la resposta. En l'exemple s'observa que completa la resolució dels noms de màquina que hi ha a la secció *asnwer* tot indicant la seva adreça IP corresponent.

2.2.4 Evolució del protocol DNS i qüestions de seguretat

Els protocols DNS han evolucionat molt gràcies a les noves necessitats provocades per l'increment exponencial de les xarxes. Una de les principals vies en evolució és el DDNS o dinàmic DNS i el DNSSEC o DNS segur.

DDNS

El protocol DDNS (*dynamic DNS*) permet que les dades del servidor DNS s'actualitzin en temps real. El principal ús és permetre que clients amb adreces IP dinàmiques d'interval puguin disposar d'un nom de domini (a pesar que la seva adreça IP varia d'una sessió a una altra). Un mecanisme consisteix a permetre que els servidors DHCP es comuniquin amb els servidors DNS i els notifiquin les actualitzacions a la base de dades de DNS que cal fer. En funció de les configuracions de xarxa que els servidors DHCP proporcionen als seus clients es fan les actualitzacions al DNS.

DNSSEC

DNSSEC o *Domain Name System Security Extensions* (extensions de seguretat per a DNS) són un conjunt d'especificacions de seguretat per permetre una comunicació DNS segura, de manera que el client pugui estar plenament segur que qui li respon les consultes és el seu servidor DNS i no un impostar (el *man-in-the-middle*). També garanteix la integritat de les dades tant de les consultes com de les respostes i a més a més està dissenyat per prevenir atacs de denegació de servei.

Man-in-the-middle

S'anomena man-in-the-middle aquells equips que es fan passar per altres en una connexió de xarxa. El client creu connectar amb el seu banc, però de fet està connectant amb un "atacant" situat entremig. Aquest atacant rep el trànsit del client i el transfereix al banc, rep la resposta del banc i la passa al client. En aquest procés està en disposició de "manipular" tot aquest trànsit a la seva conveniència.

Servidors DNS enverinats

Un dels principals problemes del protocol DNS (com de tots els primers protocols d'Internet) és la falta de seguretat. Va ser dissenyat en una època de ‚bon rotllo' on es confiava amb els altres integrants de la xarxa. Això avui en dia no és massa sensat.

Imagineu que un atacant aconsegueix el control d'un servidor DNS o aconsegueix fer passar el seu servidor DNS fals com a servidor d'un conjunt d'usuaris. Cada vegada que aquests clients fan una consulta a Internet, per exemple al seu banc (posant el nom de la web), el servidor DNS enverinat proporciona una adreça IP no del banc real sinó d'una web falsa amb la finalitat de...

Cada cop que accedim a una web, un servei, un *host*, etc. per mitjà del seu nom de domini, confiem que el servidor DNS proporcioni l'adreça IP correcta. Som confiats, oi?

2.3 Tipus de registres

El sistema de noms de domini és una base de dades jeràrquica i distribuïda en què cada servidor de noms gestiona la informació corresponent a la zona de la qual és autoritari. Cada zona conté informació dels*hosts* que la formen. La informació de zona s'emmagatzema en forma de **registre de recurs o resource record (RR)**. Hi ha la informació que permet identificar cada nom de domini amb l'adreça IP corresponent, anomenat **forward mapping o resolució directa**. També conté la informació per identificar cada adreça IP amb el nom de domini corresponent, anomenat **reverse mapping o resolució inversa**. La informació de zones conté altres informacions que permeten identificar els servidors DNS autoritaris per la zona, els servidors de correu, etc.

2.3.1 Base de dades de zona

Aprofitar fitxers de zona

Els fitxers de zona de descripció del *loopback* i dels nodes arrel són pràcticament iguals per a totes les zones, de manera que usualment es copien d'una zona ja existent en lloc d'escriure'ls de nou.

La configuració d'una zona s'emmagatzema en un conjunt de fitxers anomenat *fitxers de zona*. L'especificació del DNS diu com han de ser aquests fitxers de zona i com s'hi han de descriure els registres de recurs (descripció de cada element que pertany a la zona).

> El conjunt dels registres de recurs de totes les zones de l'espai de noms formen la **base de dades** distribuïda jeràrquica del sistema DNS.

En qualsevol zona hi haurà almenys els **fitxers de zona** següents:

- Un fitxer amb les associacions dels noms de domini a adreces IP. Aquest fitxer defineix la resolució directa.
- Un fitxer per a cada subxarxa amb l'associació de cada adreça IP al seu nom de domini canònic. Defineix la resolució inversa.
- Un fitxer amb la definició de la resolució inversa del *loopback*.
- Un fitxer amb la descripció dels nodes arrel d'Internet.

Aplicacions DNS

Hi ha diverses aplicacions que proporcionen el servei de servidor de noms. La més famosa, estesa i utilitzada és el BIND (Berkleley I N D). En la versió BIND 9 s'utilitza un fitxer de configuració anomenat */etc/named.conf* per configurar el servidor i indicar-li quins són i on es troben els fitxers de zona.

Un cop els fitxers de zona contenen tots els registres de recurs necessaris cal configurar el servidor de noms perquè utilitzi aquests fitxers. Si bé la configuració dels fitxers de zona és estàndard (definida per l'especificació DNS), la configuració del servidor depèn del programa que s'utilitzi.

2.3.2 Registres de recurs

Cada fitxer de zona conté un conjunt d'entrades que comencen en la primera columna i cada una defineix un **registre de recurs (RR)**. Els més usuals són *SOA*, *NS*, *A*, *CNAME*, *PTR* i *MX*. L'ordre en què apareixen és indiferent, però usualment és el mateix que apareix en els exemples. Cada línia té el format:

```
1 domini classe [ttl] tipus rdata:
```

- **domini** és el nom de domini que s'està definint,
- **classe**, actualment, només pren el valor *IN* per Internet.
- **ttl** és un camp opcional que descriu el temps de vida que cal emmagatzemar aquest registre en *cache*.
- **tipus** és el tipus d'RR que s'està definint.
- **rdata** és el valor que s'associa al nom de domini que es defineix.

Tot i que en cada registre de recurs es pot definir un TTL (**time to live** o temps de vida propi), el més usual és definir un TTL genèric per a totes les entrades del fitxer de zona en lloc de fer-ho una a una. El servidor BIND 9 utilitza la directiva *$TTL* (per exemple: $ttl1 1h) per indicar el temps que els altres servidors de noms han de guardar en la seva *cache* les respostes d'aquest servidor (una hora en l'exemple).

En la secció "Annexos" del web d'aquest mòdul trobareu altres tipus de recursos RR. També es poden trobar en l'especificació DNS.

Registre SOA

El registre de recurs **SOA** o **start of authority** (inici de definició de zona amb autoritat) diu que el fitxer de zona on es troba és la millor font de dades per a la zona, que el servidor de noms és autoritari per a la zona. Acostuma a ser el primer RR que hi ha en el fitxer de zona, tot i que no és obligatori. Per cada fitxer de zona hi ha d'haver només un registre SOA.

Un registre SOA té el format:

```
1 nomDomini. IN SOA nsPrimari. admin.nsPrimari. (opcions–slaves)
```

Un exemple seria el següent:

```
1 inf.ioc.cat. IN SOA ns1.inf.ioc.cat. admin.ns1.inf.ioc.cat. ( 1h 3h 1h 1w 1h )
```

La descripció de cada camp és la següent:

Punt final en el nom de domini

Posar, o no, el punt al final d'un nom de domini és important. Si acaba amb punt és un nom de domini absolut. Si acaba sense punt, és un nom relatiu i s'hi afegirà el domini per defecte al final.

- **nomDomini.** indica el nom del domini que s'està definint i pel qual el servidor de noms és autoritari. Fixeu-vos en el punt al final del nom del domini, és important posar-lo.
- **IN** indica que la classe és Internet.

- **SOA** descriu que és un registre de recurs tipus SOA.
- **nsPrimari.** és el nom del *host* servidor de noms primari per a aquesta zona. Un altre cop pareu atenció al punt del final.
- **admin.nsPrimari.** és l'adreça de correu electrònic de l'administrador del servidor de noms de domini, amb el format *usuari.servidor*. El primer punt que separa el nom d'usuari i el nom del servidor cal interpretar-lo com si fos una @ (*usuari@servidor*).
- **opcions-slaves** són paràmetres que s'indiquen entre parèntesis i que serveixen per definir com ha de ser la comunicació entre el servidor primari (o *master*) i els servidors secundaris (o *slaves*). A grans trets s'indiquen els conceptes següents:
  - *Serial*: el número de sèrie de la versió de les dades. A cada canvi de les dades de la zona, el número s'incrementa.
  - *Refresh*: temps a transcórrer entre cada refresc de dades del servidor secundari.
  - *Retry*: temps d'espera per tornar a intentar un refresc si el servidor secundari ha fallat en l'intent d'actualitzar les seves dades del servidor primari.
  - *Expire*: temps a partir del qual les dades del servidor secundari es consideren sense autoritat si no s'han refrescat abans.
  - *Minimum*: valor del TTL dels camps per defecte. Recordeu que a cada camp s'hi pot assignar un TTL específic. Segons la versió del servidor indicarà el TTL de les respostes negatives (*negative caching*), ja que el temps TTL es defineix per la directiva $TTL.

Registre NS

El registre de recurs **NS** o name server (**servidor de noms**) defineix un servidor de noms autoritatiu per a la zona. Hi haurà tantes entrades NS com servidors de noms autoritatius hi ha en la zona. L'estàndard DNS en recomana almenys dos (un de primari o *master* i un de seguretat secundari o *slave*).

Un registre NS consta dels camps:

```
1 nomDomini. IN NS nameServer.
```

Un exemple seria aquest:

```
1 inf.ioc.cat. IN NS ns1.inf.ioc.cat.
```

La descripció de cada camp és la següent:

- **nomDomini.** indica el nom del domini que s'està definint.
- **IN** indica que la classe és Internet.

- **NS** descriu que es tracta d'un tipus de registre de recurs en què es defineix un servidor de noms.
- **nameServer.** és el nom del servidor de noms. Fixeu-vos un altre cop que tant *nomDomini.* com *nameServer.* acaben en un punt per indicar que són noms de domini absoluts o FQDN.

En la següent llista es pot veure part d'una resposta a una consulta nslookup per observar quins són els servidors de noms de *yahoo*:

```
... output suprimit ...
Authoritative answers can be found from:
yahoo.com       nameserver = ns2.yahoo.com.
yahoo.com       nameserver = ns1.yahoo.com.
yahoo.com       nameserver = ns3.yahoo.com.
yahoo.com       nameserver = ns4.yahoo.com.
yahoo.com       nameserver = ns5.yahoo.com.
yahoo.com       nameserver = ns6.yahoo.com.
yahoo.com       nameserver = ns8.yahoo.com.
ns2.yahoo.com   internet address = 68.142.255.16
ns1.yahoo.com   internet address = 66.218.71.63
ns3.yahoo.com   internet address = 217.12.4.104
ns4.yahoo.com   internet address = 68.142.196.63
ns5.yahoo.com   internet address = 216.109.116.17
ns6.yahoo.com   internet address = 202.43.223.170
ns8.yahoo.com   internet address = 202.165.104.22
```

Registre A

Un registre de recurs **A** o address **(adreça)** associa un nom de *host* a una adreça IP (resolució directa). Per cada nom de *host* de la xarxa caldrà disposar d'una entrada on s'associï el nom del *host* a la seva adreça IP.

Un registre A consta dels camps:

```
nomHost. IN A IP
```

Un exemple seria aquest:

```
mahatma.inf.ioc.cat. IN A 192.168.0.2
```

La descripció de cada camp és la següent:

- **nomHost.** indica el nom del *host* que s'està definint.
- **IN** indica que la classe és Internet.
- **A** descriu que es tracta d'un tipus de registre de recurs de definició d'adreça IP.
- **IP** és l'adreça IP assignada al *host*.

Fixem-nos un altre cop que *nomHost.* acaba en punt per indicar el seu FQDN. Un *host* pot tenir més d'una IP assignada al mateix nom de *host* i s'anomena multi-homed. Simplement caldrà que hi hagi un registre A per a cada adreça IP. Constarà

del mateix nom de *host* a l'esquerra de la definició, i la corresponent adreça IP a la dreta. Per exemple:

A vs CNAME

Compte! No s'ha de confondre els registres de recurs *A* i els *CNAME*:*hosts* registre A àlies registre CNAME

```
superserver.dom.com. IN A 10.0.0.1
superserver.dom.com. IN A 10.0.0.2
```

Els noms definits en els registres de tipus A són noms **canònics**. Un *host* es pot identificar per més d'un nom, però només un és el nom canònic (original), la resta són **àlies**. Els noms canònics es defineixen amb el tipus de registre A. Els àlies es defineixen amb el tipus de registre CNAME.

Registre CNAME

Els registres de recurs **CNAME** o canonical name **(nom canònic)** associen un àlies a un nom canònic.

Un registre CNAME consta dels camps:

```
nomHost. IN CNAME hostCanonicalName. | IP
```

Un exemple seria aquest:

```
ftp.inf.ioc.cat. IN CNAME mahatma.inf.ioc.cat.
tftp.inf.ioc.cat IN CNAME 192.168.0.2
```

La descripció de cada camp és la següent:

- **nomHost.** indica el nom de l'àlies que s'està definint.
- **IN** indica que la classe és Internet.
- **CNAME** descriu que es tracta d'un registre de recurs de definició d'un àlies.
- **hostCanonicalName | IP** és el nom de *host* canònic al qual s'assigna l'àlies. Fixeu-vos un altre cop que és un FQDN i acaba en punt. Generalment, els registres CNAME tenen a la part dreta de la definició un nom canònic, però de vegades caldrà indicar-hi una adreça IP. Penseu en un*host multi-homed* amb múltiples adreces IP que a més a més té àlies. Si la definició fos pel nom canònic del *host*, no se sabria quina de les adreces IP correspon a l'àlies. En aquests casos, el CNAME apunta a una adreça IP.

Exemple de host multi-homed

Es vol posar l'àlies *super1* i *super2* a cada una de les IP del host superserver.com (un *host* que té dues adreces IP assignades a aquest nom). Les entrades CNAME serien les següents:

- super1.dom.com. IN CNAME 10.0.0.1.
- super2.dom.com. IN CNAME 10.0.0.2.

La resolució dels àlies s'obté buscant l'entrada de l'àlies en el fitxer de zona. Amb l'entrada CNAME s'obté el nom canònic corresponent a l'àlies. Un altre cop es torna a buscar en el fitxer de zona, ara el nom canònic. Una entrada de tipus A proporcionarà l'adreça IP corresponent (àlies –> CNAME –> nom canònic –> A –> adreça IP). Un àlies mai pot aparèixer a la part dreta d'una definició de registre de recurs.

Registre PTR

Un registre de recurs **PTR** o pointer **(punter)** associa una adreça IP al nom de *host* pertinent (resolució inversa). Cal una entrada PTR per a cada interfície de xarxa de la zona.

Un registre PTR consta dels camps:

```
ipInversa.in–addr.arpa. PTR hostCanonicalName.
```

Un exemple seria aquest:

```
2.20.168.192.in–addr.apra. IN PTR mahatma.inf.ioc.cat.
```

La descripció de cada camp és la següent:

- **ipInversa.in-addr.arpa.** indica l'adreça IP escrita en forma de domini*in-addr.arpa* per poder fer la resolució inversa. Les adreces IP s'escriuen al revés quan formen part del domini *in-addr.arpa*. Així una IP 192.168.20.2 s'escriu *2.20.168.192.in-addr.arpa.*
- **IN** indica que la classe és Internet.
- **PTR** descriu que es tracta d'un registre de recurs de definició de la resolució inversa d'una adreça IP.
- **hostCanonicalName.** és el nom de *host* FQDN assignat a l'adreça IP. El nom del *host* ha de ser per força el nom canònic. No hi pot haver dues definicions de la mateixa IP amb noms diferents (àlies), només de la IP al nom canònic.

Registre MX

Un registre **MX** mail echanger **(servidor de correu electrònic)** defineix un servidor de correu. Es pot posar una entrada MX per a cada servidor de correu, però no és obligatori que n'hi hagi cap.

Un registre MX consta dels camps:

```
nomDomini. IN MX num mailServer.
```

Un exemple seria aquest:

```
inf.ioc.cat. IN MX 10 ns1.inf.ioc.cat.
```

La descripció de cada camp és la següent:

- **nomDomini.** indica el nom del domini que s'està definint.
- **IN** indica que la classe és Internet.

- **MX** descriu que es tracta d'un registre de recurs on es defineix un servidor de correu per a aquest domini.
- **num** és un valor numèric que expressa el grau de preferència d'aquest servidor de correu respecte a altres servidors de correu del domini. El valor més baix és el que es prefereix més. Són valors arbitraris que defineix l'administrador de xarxes.
- **mailServer.** correspon al nom FQDN del servidor de correu que s'està definint.

Podem observar la llista de servidors de correu de *google* fent:

```
[root@localhost ~]# host google.com
google.com has address 74.125.45.100
google.com has address 74.125.127.100
google.com has address 74.125.67.100
google.com mail is handled by 10 google.com.s9b1.psmtp.com.
google.com mail is handled by 100 smtp2.google.com.
google.com mail is handled by 10 google.com.s9a2.psmtp.com.
google.com mail is handled by 10 google.com.s9b2.psmtp.com.
google.com mail is handled by 10 google.com.s9a1.psmtp.com.
google.com mail is handled by 100 smtp1.google.com.
```

Les dues llistes següents mostren exemples dels fitxers de configuració per a la resolució directa i la resolució inversa de la zona *ioc.cat*. En el primer es defineixen dos servidors de nom, un encaminador, una impressora i dos *hosts*. El primer dels servidors de noms també fa les funcions de servidor de correu, web i FTP.

```
;Exemple de fitxer de zona ioc.cat
$TTL 3D
ioc.cat. IN SOA ns1.ioc.cat. admin.ioc.cat. {
        23;serial
        8H;refresh
        2H;retry
        4W;expire
        1D);minimum ttl
      NS ns1.ioc.cat.
      NS ns2.ioc.cat.
      MX 10 correu.ioc.cat.
ns1.ioc.cat. A 192.168.0.5; servidor amb 2 ip
             A 172.16.20.5
ns2.ioc.cat. A 192.168.0.7; servidor dns slave
router       A 192.168.0.1; router. Nom relatiu
correu   CNAME ns1 ; alias correu
www      CNAME ns1 ; alias web
ftp      CNAME ns1 ; alias ftp
hp-7200c A 192.168.0.2; impressora
pc01     A 192.168.0.50
pc02     A 192.168.0.51
```

En la llista següent es pot veure com es defineix una entrada PTR per a cada nom canònic definit en la resolució directa per a una subxarxa concreta. La subxarxa *192.168.0.0/24* utilitza el fitxer *0.168.192.in-addr.arpa*.

```
; Zona 0.168.192.in-addr.arpa.
;Exemple de fitxer de zona inversa ioc.cat
; correspon a la xarxa 192.168.0.0/24
$TTL 3D
```

```
ioc.cat. IN SOA ns1.ioc.cat. admin.ioc.cat. {
     23; serial
     8H; refresh
     2H; retry
     4W; expire
     1D) ; minimum ttl
    NS ns1.ioc.cat.
5   IN PTR ns1.ioc.cat.
7   IN PTR ns2.ioc.cat.
1   IN PTR router.ioc.cat.
2   IN PTR hp-7200c.ioc.cat.
50  IN PTR pc01.ioc.cat.
51  IN PTR pc02.ioc.cat.
```

2.3.3 Altres registres

Hi ha altres tipus de registres de recurs que no són tan utilitzats i que es mencionen a continuació:

- HINFO: (Host Information) informació sobre el tipus d'ordinador.
- MB: (Mail Box) informació sobre una bústia de correu.
- MG: (Mailgroup) informació sobre un grup de correu.
- MR: nom nou d'una bústia de correu.
- WKS (Well Known Services) llista de serveis del *host*.
- TXT: (Text) text descriptiu.
- NULL: (Null) registre buit.
- AAAA () corresponent a una adreça de host usant Ipv6.

2.3.4 Abreviacions

L'estàndard DNS permet fer abreviacions en els fitxers de definició de zona per tal de facilitar-ne la sintaxi. Les més importants són:

- Es pot usar @ com a indicador del nom de domini quan és el mateix que el nom de domini origen (el que s'està definint).
- Si no s'indica un nom de domini en el primer camp i es deixa buit, s'entén el mateix nom que el definit en el registre anterior.
- Als noms de domini relatius (no acabats en punt) se'ls afegeix el nom de domini origen o nom de la zona que s'està definint.

Un exemple amb abreviacions: zona inf.ioc.cat:

```
@ IN SOA ns1.inf.ioc.cat. Admin.ns1.ioc.cat (1 3h 1h 1w 1h)
                        ;utilitzat @ per referir–se al domini inf.ioc.cat
IN NS ns1     ;primera columna pren idem valor que l'anterior @
IN NS ns2     ;primera columna pren idem valor que l'anterior, @
mahatma IN A 192.168.0.2
ftp IN CNAME mahatma       ;als noms relatius s'afegeix inf.ioc.cat
```

2.4 Instal·lació d'un servei DNS

El servei de xarxa DNS està estructurat en forma de servei client/servidor; per tant, caldrà disposar del programari apropiat per fer cada un d'aquests rols. El programari que fa la funció de client usualment ja està integrat en el sistema operatiu (la part que gestiona la xarxa) o en les mateixes aplicacions (per exemple el *Firefox*). És a dir, per disposar de la part client del servei DNS normalment no cal instal·lar res, tot i que si cal configurar-lo correctament.

Així doncs, quan parlem d'instal·lar un servei DNS fem referència al procés d'instal·lació i configuració del programari del servidor. Evidentment també caldrà configurar els clients adequadament per fer ús d'aquest servei.

La instal·lació del programari que proporciona el servei DNS es fa de manera molt similar (per no dir idèntica) al d'altres serveis de xarxa com els serveis DHCP, HTTP, FTP, etc. Es tracta d'instal·lar el programari de l'aplicació servidor i fer-ne la configuració apropiada. Senzill oi?

Per fer això cal plantejar-se els següents passos:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Obtenir l'aplicació que proporciona el servei DNS.
- Observar l'estat de la xarxa actual. Està el servei ja en funcionament? Existeix ja una configuració DNS activa?
- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i activar els clients perquè la utilitzin.
- Comprovar que el servei funciona correctament.

2.4.1 Aplicacions servidor DNS

Sempre que l'administrador vol posar en funcionament un nou servei de xarxa cal que primerament analitzi quines aplicacions hi ha al mercat que ofereixen aquest

servei. És feina seva estudiar les característiques de les diverses aplicacions, com per exemple: avaluar-ne l'eficiència, el cost, el que en diuen els altres... La manera més fàcil de fer això és navegar per Internet, consultar les revistes especialitzades o demanar consell a un dels gurus informàtics coneguts.

Cerca de DNS a Internet

Usualment l'administrador s'informa a través del seu cercador preferit, per exemple *Google*, i de webs com la *viquipèdia*. Proveu a buscar DNS o DNS server al *Google* i a la *wiki* (en anglès).

Usualment, però, l'administrador acaba utilitzant l'aplicació servidor DNS que li proporciona el mateix sistema operatiu. Si utilitzeu el sistema operatiu Windows l'empresa Microsoft disposa d'una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment si utilitzeu GNU/Linux, segurament la mateixa distribució ja proporciona un servidor DNS o bé n'existeix algun de clàssic provinent de l'*Unix*. De totes maneres en podeu obtenir d'altres també a Internet. En la figura 2.2 es pot veure quin servidor utilitzen els nodes arrels.

FIGURA 2.2. Llista de servidors arrel DNS i programari que utilitzen

Letter	IPv4 address	IPv6 address	Old name	Operator	Location	Software
A	198.41.0.4	2001:503:BA3E::2:30	ns.internic.net	VeriSign	distributed using anycast	BIND
B	192.228.79.201	2001:478:65::53	ns1.isi.edu	USC-ISI	Marina Del Rey, California, U.S.	BIND
C	192.33.4.12		c.psi.net	Cogent Communications	distributed using anycast	BIND
D	128.8.10.90		terp.umd.edu	University of Maryland	College Park, Maryland, U.S.	BIND
E	192.203.230.10		ns.nasa.gov	NASA	Mountain View, California, U.S.	BIND
F	192.5.5.241	2001:500:2f::f	ns.isc.org	Internet Systems Consortium	distributed using anycast	BIND 9[3]
G	192.112.36.4		ns.nic.ddn.mil	Defense Information Systems Agency	distributed using anycast	BIND
H	128.63.2.53	2001:500:1::803f:235	aos.arl.army.mil	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, U.S.	NSD
I	192.36.148.17	2001:7fe::53 (testing)	nic.nordu.net	Autonomica	distributed using anycast	BIND
J	192.58.128.30	2001:503:C27::2:30		VeriSign	distributed using anycast	BIND
K	193.0.14.129	2001:7fd::1		RIPE NCC	distributed using anycast	NSD[4]
L	199.7.83.42 (since November 2007; originally was 198.32.64.12)[5]	2001:500:3::42		ICANN	distributed using anycast	NSD[6]
M	202.12.27.33	2001:dc3::35		WIDE Project	distributed using anycast	BIND

Llista extreta de la wikipedia on es poden observar els 13 servidors o nodes arrel del servei DNS, l'entitat que els gestiona, el tipus de difusió que fan i el programari que utilitzen.

2.4.2 Instal·lar l'aplicació servidor

Tot seguit es descriurà el procés per instal·lar el servei DNS en un entorn GNU/Linux. Un cop feta la instal·lació cal observar què s'ha instal·lat, quins programes executables, on són els fitxers de configuració, els de monitoratge, etc.

Els usuaris GNU/Linux poden buscar fàcilment per Internet quins paquets del client i del servidor dhcp usant eines com *yum*, *apt-get* o *wget*, a part dels repositoris de programari usuals o del mateix *Google*.

Buscar per Internet paquets del client i del servidor DNS. A *Google*, a repositoris de programari, etc. Si es disposa de yum o de apt-get o wget:

Llista de paquets *rpm* que contenen el text *bind*:

```
[root@portatil ~]# yum list bind
Installed Packages
bind.i386                                31:9.4.2-3.fc7         installed
Available Packages
bind.i386                                31:9.4.2-4.fc7         updates
```

Llista de paquets que contenen *bind**:

```
[root@portatil ~]# yum list bind*
Installed Packages
bind.i386                                31:9.4.2-3.fc7         installed
bind-libs.i386                           31:9.4.2-3.fc7         installed
bind-utils.i386                          31:9.4.2-3.fc7         installed
Available Packages
bind.i386                                31:9.4.2-4.fc7         updates
bind-chroot.i386                         31:9.4.2-4.fc7         updates
bind-devel.i386                          31:9.4.2-4.fc7         updates
bind-libs.i386                           31:9.4.2-4.fc7         updates
bind-sdb.i386                            31:9.4.2-4.fc7         updates
bind-utils.i386                          31:9.4.2-4.fc7         updates
```

Instal·lar el paquet *bind*:

```
# yum install bind
```

Fer la llista dels paquets *bind* instal·lats. Si el sistema ja els té instal·lats o volem comprovar-ho podem consultar els paquets instal·lats:

```
[root@portatil ~]# rpm -qa | grep bind
ypbind-1.19-9.fc7
system-config-bind-4.0.2-6.fc7
rpcbind-0.1.4-8.fc7
bind-libs-9.4.2-3.fc7
bind-9.4.2-3.fc7
bind-utils-9.4.2-3.fc7
```

Obtenir informació del paquet del servei *bind*:

```
[root@portatil ~]# rpm -qi bind
Name        : bind                         Relocations: (not relocatable)
Version     : 9.4.2                             Vendor: Fedora Project
Release     : 3.fc7                      Build Date: dl 21 gen 2008 11:27:32
     CET
Install Date: dc 23 gen 2008 19:12:18 CET      Build Host: xenbuilder4.fedora.
    phx.redhat.com
Group       : System Environment/Daemons    Source RPM: bind-9.4.2-3.fc7.src.
    rpm
Size        : 3627903                          License: BSD-like
Signature   : DSA/SHA1, dl 21 gen 2008 19:48:26 CET, Key ID b44269d04f2a6fd2
Packager    : Fedora Project
URL         : http://www.isc.org/products/BIND/
Summary     : El servidor de noms de domini (DNS) Berkley Internet Name Domain
    (BIND).
Description :
El BIND (Berkeley Internet Name Domain) és una implementació de protocol
DNS (sistema de noms de domini). El BIND inclou un servidor DNS (named),
que converteix els noms d'ordinador en adreces IP, una biblioteca per resoldre
    els noms
(rutines per aplicacions que usen DNS), i eines per
a verificar que el servidor DNS funciona degudament.
```

2.4.3 Observar els components del paquet

Fer la llista dels components del paquet *bind*:

```
[root@portatil ~]# rpm -ql bind
/etc/dbus-1/system.d/named.conf
/etc/logrotate.d/named
/etc/named.conf
/etc/rc.d/init.d/named
/etc/rndc.conf
/etc/rndc.key
/etc/sysconfig/named
/usr/sbin/dns-keygen
/usr/sbin/dnssec-keygen
/usr/sbin/dnssec-signzone
/usr/sbin/lwresd
/usr/sbin/named
/usr/sbin/named-bootconf
/usr/sbin/named-checkconf
/usr/sbin/named-checkzone
/usr/sbin/named-compilezone
/usr/sbin/namedGetForwarders
/usr/sbin/namedSetForwarders
/usr/sbin/rndc
/usr/sbin/rndc-confgen
/usr/share/dbus-1/services/named.service
... output suprimit ...
/usr/share/doc/bind-9.4.2/sample/var/named/slaves/
my.slave.internal.zone.db
/usr/share/man/man5/named.conf.5.gz
... output suprimit ...
/usr/share/man/man8/rndc.8.gz
/var/named
/var/named/data
/var/named/dynamic
/var/named/slaves
/var/run/named
```

En funció del directori on s'ubiquen podem intuir si són executables, de configuració o de documentació. També podem mirar de filtrar la sortida en cada cas:

Fitxers de configuració:

```
[root@portatil ~]# rpm -qc bind
/etc/dbus-1/system.d/named.conf
/etc/logrotate.d/named
/etc/named.conf
/etc/rc.d/init.d/named
/etc/rndc.conf
/etc/rndc.key
/etc/sysconfig/named
/usr/share/dbus-1/services/named.service
[root@portatil ~]# rpm -ql  bind  | grep etc
/etc/dbus-1/system.d/named.conf
/etc/logrotate.d/named
/etc/named.conf
/etc/rc.d/init.d/named
/etc/rndc.conf
/etc/rndc.key
/etc/sysconfig/named
/usr/share/doc/bind-9.4.2/sample/etc
/usr/share/doc/bind-9.4.2/sample/etc/named.conf
/usr/share/doc/bind-9.4.2/sample/etc/rndc.conf
```

Fitxers de documentació:

```
[root@portatil ~]# rpm -qd bind
/usr/share/doc/bind-9.4.2/CHANGES
/usr/share/doc/bind-9.4.2/COPYRIGHT
/usr/share/doc/bind-9.4.2/README
/usr/share/doc/bind-9.4.2/README.DBUS
/usr/share/doc/bind-9.4.2/arm/Bv9ARM-book.xml
... output suprimit ...
/usr/share/doc/bind-9.4.2/misc/sdb
/usr/share/doc/bind-9.4.2/sample/etc/named.conf
/usr/share/doc/bind-9.4.2/sample/etc/rndc.conf
/usr/share/doc/bind-9.4.2/sample/named.ca
/usr/share/doc/bind-9.4.2/sample/named.empty
/usr/share/doc/bind-9.4.2/sample/named.localhost
/usr/share/doc/bind-9.4.2/sample/named.loopback
/usr/share/doc/bind-9.4.2/sample/named.rfc1912.zones
/usr/share/doc/bind-9.4.2/sample/var/named/my.external.zone.db
/usr/share/doc/bind-9.4.2/sample/var/named/my.internal.zone.db
/usr/share/doc/bind-9.4.2/sample/var/named/slaves/my.ddns.internal.zone.db
/usr/share/doc/bind-9.4.2/sample/var/named/slaves/my.slave.internal.zone.db
/usr/share/man/man5/named.conf.5.gz
/usr/share/man/man5/rndc.conf.5.gz
/usr/share/man/man8/dnssec-keygen.8.gz
/usr/share/man/man8/dnssec-signzone.8.gz
/usr/share/man/man8/lwresd.8.gz
/usr/share/man/man8/named-checkconf.8.gz
/usr/share/man/man8/named-checkzone.8.gz
/usr/share/man/man8/named-compilezone.8.gz
/usr/share/man/man8/named.8.gz
/usr/share/man/man8/rndc-confgen.8.gz
/usr/share/man/man8/rndc.8.gz
```

Podem mirar de filtrar quins són els executables tenint en compte que usualment estaran en un directori de nom *bin* o *sbin*:

```
[root@portatil ~]# rpm -ql bind   | grep bin/
/usr/sbin/dns-keygen
/usr/sbin/dnssec-keygen
/usr/sbin/dnssec-signzone
/usr/sbin/lwresd
/usr/sbin/named
/usr/sbin/named-bootconf
/usr/sbin/named-checkconf
/usr/sbin/named-checkzone
/usr/sbin/named-compilezone
/usr/sbin/namedGetForwarders
/usr/sbin/namedSetForwarders
/usr/sbin/rndc
/usr/sbin/rndc-confgen
```

En resum:

- Els fitxers de documentació es troben generalment a: */usr/share/doc* i a */usr/share/man*.
- Els fitxers de configuració es troben a: */etc*, */etc/sysconfig*.
- El dimoni del servei es troba a: */usr/sbin/named*.
- El fitxer de configuració del dimoni del servei *vsftpd* es: */etc/named.conf*.
- El fitxer de govern del servei és: */etc/rc.d/init.d/named*.
- El fitxer de configuració del registre de *logs* es troba a */etc/logrotate.d/named*.

Ubicació de fitxers

En GNU/Linux els fitxers executables per l'administrador es troben usualment a */sbin* i a */usr/sbin*. Els executables d'usuari normalment són a */bin* i */usr/bin*.

2.4.4 Activar/desactivar el servei i establir els nivells d'arrencada

Un cop s'ha instal·lat al sistema un servei de xarxa, cal posar-lo en funcionament. Primer caldrà determinar quin tipus de servei és, si autònom o integrat, dins del superservei de xarxa. Un cop fet això, cal saber si ja està en funcionament o no. De fet cal saber engegar-lo, aturar-lo i reinicialitzarlo. Finalment, cal establir quin estat ha de tenir el servei per defecte cada cop que s'engegui el servidor.

El servei

Primerament cal saber si el servidor instal·lat funciona "*stand-alone*" o dins del superdimoni de xarxa "*xinetd*" o "*initd*". Si existeixen fitxers de configuració dins del directori */etc/xinetd.d/<nom-servei>* es tracta d'un servei dins del *xinetd*. Si existeixen fitxers de configuració dins del directori */etc/rc.d/init.d/<nom-servei>* es tracta d'un servei "*stand-alone*".

Observem ara el contingut del paquet per intentar saber si els fitxers de configuració ens permeten saber de quin tipus de servei es tracta:

```
[root@portatil ~]# rpm -ql bind   | grep /etc
/etc/dbus-1/system.d/named.conf
/etc/logrotate.d/named
/etc/named.conf
/etc/rc.d/init.d/named
/etc/rndc.conf
/etc/rndc.key
/etc/sysconfig/named
/usr/share/doc/bind-9.4.2/sample/etc
/usr/share/doc/bind-9.4.2/sample/etc/named.conf
/usr/share/doc/bind-9.4.2/sample/etc/rndc.conf
```

Com podem observar es tracta d'un servei *stand-alone*. També es pot consultar el tipus de servei amb l'ordre chkconfig i observar si surt la llista d'un tipus o de l'altre:

```
[root@portatil ~]# chkconfig --list | grep named
named           0:apagat        1:apagat    2:apagat
3:apagat        4:apagat        5:apagat    6:apagat
```

Per facilitar buscar els serveis *stand-alone* podem fer:

```
[root@portatil ~]# chkconfig --list named
named           0:apagat        1:apagat    2:apagat
3:apagat        4:apagat        5:apagat    6:apagat
```

Estat del servei

Es pot saber l'estat del servei amb l'opció *status* de les ordres:

```
[root@portatil ~]# service named   status
named   està aturat
[root@portatil ~]# /etc/rc.d/init.d/named status
named   està aturat
```

Es pot arrencar el servei amb l'opció *start* de les ordres:

```
[root@portatil ~]# service named   start
S'està iniciant el servei named:                    [ FET  ]
[root@portatil ~]# /etc/rc.d/init.d/named start
S'està iniciant el servei named:                    [ FET  ]
```

Es pot aturar el servei amb l'opció *stop* de les ordres:

```
[root@portatil ~]# service named   stop
S'està aturant el named:                            [ FET  ]
[root@portatil ~]# /etc/rc.d/init.d/named stop
S'està aturant el servei named:                     [ FET  ]
```

Es pot iniciar de nou el servei (recarregar) amb l'opció *reload* o *restart* de les ordres:

```
[root@portatil ~]# /etc/rc.d/init.d/named restart
S'està aturant el servei named:                   [Incorrecte]
S'està iniciant el servei named:                    [ FET  ]
[root@portatil ~]# service named  reload
S'està aturant el servei named:                     [ FET  ]
S'està iniciant el servei named:                    [ FET  ]
```

Per saber les ordres possibles:

```
[root@portatil ~]# service named   patapum
Forma d'ús: /etc/init.d/dhcpd {start|stop|restart|condrestart|status}
[root@portatil ~]# /etc/rc.d/init.d/named pimpam
Forma d'ús: /etc/rc.d/init.d/dhcpd {start|stop|restart|condrestart|status}
```

Establir els nivells per defecte del servei

Els serveis (els dimonis executables) es poden configurar per arrencar automàticament en determinats nivells d'execució. Les màquines GNU/Linux tenen 7 nivells d'execució com es pot veure del fitxer */etc/inittab*:

```
[root@portatil ~]# head -20 /etc/inittab
... output suprimit ...
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
# ... output suprimit ...
```

Per configurar a quins nivells es vol que s'executi un servei s'utilitza l'ordrechkconfig, que permet activar/desactivar el servei pels nivells indicats:

```
[root@portatil ~]# chkconfig --list dhcpd
dhcpd           0:apagat        1:apagat    2:apagat
3:apagat        4:apagat        5:apagat    6:apagat

[root@portatil ~]# chkconfig --help
chkconfig versió 1.3.34 - Copyright (C) 1997-2000 Red Hat, Inc.
```

```
Aquest programari es pot distribuir lliurement d'acord amb els termes de la
    Llicència Pública General GNU.
forma d'ús:   chkconfig --list [nom]
         chkconfig --add <nom>
         chkconfig --del <nom>
         chkconfig --override <nom>
         chkconfig [--level <nivells>] <nom> <on|off|reset|resetpriorities>

[root@portatil ~]# chkconfig --level 345 namedd on
[root@portatil ~]# chkconfig --list | grep named
named           0:apagat   1:apagat        2:apagat
3:engegat       4:engegat  5:engegat       6:apagat
```

Fixeu-vos que definir els nivells d'execució no significa que el servei estigui ara engegat. Significa que quan arranqui el sistema (a partir d'ara) s'engegarà en els nivells corresponents. Podem ara estar al nivell 5 i tenir el servei aturat perquè encara no l'hem engegat. Exemple:

```
[root@portatil ~]# runlevel
N 5
[root@portatil ~]# service named  status
named  està aturat
[root@portatil ~]# service named  start
S'està iniciant el servei named                    [ FET  ]
```

2.5 Servidor només

La configuració completa d'un servidor DNS no és que sigui difícil, però es "entretinguda", cal crear cada fitxer de zona amb les entrades pertinents de cada *host* de la zona. Hi ha organitzacions petites on no els cal configurar el servei DNS completament, en tindrien prou a tenir un servidor DNS local que permetés accelerar les consultes DNS que es fan a l'exterior.

Sabem que tota resolució de nom de domini comporta una consulta a un servidor DNS, usualment el del proveïdor de servei (ISP). Es pot posar en marxa un servidor només *cache* que simplement a una xarxa local per tal de proporcionar més eficiència a les consultes. El servidor només *cache* no administra cap zona, no té registres de recurs, simplement rep les consultes dels clients, les tramet al servidor DNS extern, rep les seves respostes, les desa al *cache* i les retorna al client.

El benefici d'aquest esquema és que el servidor només cache acumula en memòria (a la *cache*) les respostes que va obtenint. En les consultes següents, si es demana per als mateixos dominis, ja no li cal passar la consulta a l'exterior sinó simplement respondre de la *cache*. Evidentment les seves respostes són sempre no autoritàries.

Un servidor **només** cache només emmagatzema les respostes d'altres servidors externs a memòria, però no gestiona cap zona. No és autoritari, simplement augmenta l'eficiència quan rep consultes de les quals ja sap la resposta (les té a la *cache*).

Tot seguit es mostra un exemple de configuració d'un servidor només *cache*:

```
#Exemple de configuració de servidor només cache:
#Extret de la documentació de bind9.
//A Caching–only Name Server
//The following sample configuration is appropriate for a caching–only name server for use
//by clients internal to a corporation. All queries from outside clients are refused using
//the allow–query option. Alternatively, the same effect could be achieved using suitable
//firewall rules.

// Two corporate subnets we wish to allow queries from.
acl corpnets { 192.168.4.0/24; 192.168.7.0/24; };

options {
    directory "/etc/namedb"; // Working directory
    allow–query { corpnets; };
};

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in–addr.arpa" {
    type master;
    file "localhost.rev";
    notify no;
};
```

En l'exemple anterior es pot observar que no es defineix cap fitxer de zona per ser administrada, excepte el fitxer de resolució inversa del *loopback*, que cal indicar-lo sempre. Per tant, aquest servidor únicament fa la funció d'atendre peticions DNS com a intermediari i desar-les a la memòria *cache*. El directori on emmagatzema temporalment la informació és el */etc/namedb*.

Encara sense entrar a la configuració dels registres de recurs, simplement a través del fitxer de configuració general del servei, es pot configurar un servidor DNS com a secundari. Així actua com a servidor autoritari d'una zona rebent la informació a través de transferències del servidor primari, sense que calgui escriure els fitxers de configuració dels registres de recurs.

Es pot configurar fàcilment un **servidor secundari** autoritari, simplement indicant la zona que administra i de quin servidor(s) primari rebrà la transferència de zona.

Un servidor es pot configurar per no fer la funció de *cache* i treballar únicament com a servidor autoritari. Un servidor pot ser primari per a una zona i secundari per a una altra. L'exemple següent mostra una combinació d'aquests elements:

```
#Exemple de configuració de servidor només autoritatiu:
#Extret de la documentació de bind9.
//An Authoritative–only Name Server
//This sample configuration is for an authoritative–only server that is the master server
//for "example.com" and a slave for the subdomain "eng.example.com".

options {
    directory "/etc/namedb"; // Working directory
    allow–query–cache { none; }; // Do not allow access to cache
    allow–query { any; }; // This is the default
    recursion no; // Do not provide recursive service
```

```
};

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
    type master;
    file "localhost.rev";
    notify no;
};

// We are the master server for example.com
zone "example.com" {
    type master;
    file "example.com.db";
    // IP addresses of slave servers allowed to transfer example.com
    allow-transfer {
        192.168.4.14;
        192.168.5.53;
    };
};

// We are a slave server for eng.example.com
zone "eng.example.com"

type slave;
    file "eng.example.com.bk";
    // IP address of eng.example.com master server
    masters { 192.168.4.12; };
};
```

Si ens hi fixem podem observar que:

- El servidor es configura com a no *cache* (no emmagatzema respostes externes).
- Permet consultes però no recursivament (només consultes iteratives).
- És un servidor autoritari per a la zona *example.com*.
- És un servidor primari per a la zona *example.com*, com mostra l'opció*type master*.
- El domini *example.com* disposa a més a més de dos servidors secundaris (també autoritaris) que comparteixen amb aquest servidor la responsabilitat del servei DNS. Aquets dos equips són el *192.168.4.14* i*192.168.5.53*, tal com es pot observar de l'opció *allow-transfer*.
- Per tant, el domini *example.com* disposa de tres servidors de nom, un de primari i dos de secundaris o *backup*. Tots ells són autoritaris per a la zona. L'opció *allow-transfer* permet al servidor primari transferir les actualitzacions de la seva base de dades als altres dos servidors secundaris.
- La zona *eng.example.com* està delegada a un altre servidor de noms. Això es pot deduir en l'últim bloc *zone*, on es pot observar que aquesta zona realment l'administra l'equip *192.168.4.12*.
- Tot i que sembla un contrasentit, un servidor pot ser primari per a una zona i secundari per a altres. Això passa en aquest exemple. El servidor és secundari per a la zona *eng.example.com*. Es pot observar amb l'opció *type slave*.

- Per tant, per a la zona *eng.example.com* el servidor fa de *backup* o de suport. I quin és el servidor que realment conté la base de dades d'aquesta subzona? L'opció *masters* mostra que aquesta funció la fa l'equip *192.168.4.12*.

2.6 Creació d'una zona

El propòsit principal d'un servei DNS és administrar una zona, per exemple una xarxa local amb tots els equips de l'organització on es troba. Per fer això caldrà definir els fitxers de configuració del servei DNS en general (del dimoni) i definir cada una de les zones de què es compongui la xarxa. També caldrà crear els fitxers corresponents a la resolució inversa de cada xarxa i del *loopback*.

2.6.1 Configuració dels fitxers de zona

Els fitxers de zona contenen els registres de recurs que formen la base de dades de la zona. Cal configurar el servidor de noms per indicar-li quins són i on són aquests fitxers. Cada administrador anomena els fitxers com li plau o seguint l'estil marcat per l'aplicació servidor DNS que utilitza. Un exemple és anomenar els fitxers de zona amb el format *db.nomDomini* per al fitxer de resolució directa, i *db.ipSubXarxa* per a la resolució inversa per a cada xarxa de la zona. El fitxer de la zona corresponent a la resolució inversa del *loopback* es pot anomenar *db.127.0.0*, i el fitxer amb la informació dels nodes arrel es pot dir *db.cache*.

Independentment de l'aplicació que s'utilitzi com a servidor de noms, caldrà configurar-la per dir-li on són aquests fitxers, com es diuen, si fan la funció de servidor autoritari per a la zona o no, si fan la funció de primari o secundari i d'altres opcions possibles.

Exemple de configuració del servidor de noms

Si tenim una zona que es compon d'una única xarxa 192.168.20.0/ 24 amb el nom de domini inf.ioc.cat caldran cinc fitxers de zona:

- El fitxer de resolució directa *db.inf.ioc.cat.*
- El fitxer de resolució inversa *db.192.168.20.*
- El fitxer de resolució inversa del *loopback db.127.0.0.*
- El fitxer amb la informació dels nodes arrel del DNS, *db.cache.*

L'aplicació actualment més usada és l'aplicació BIND, que es configura mitjançant un fitxer anomenat /etc/named.conf. La manera d'indicar a BIND 9 quin és el directori per defecte on es troben els fitxers de configuració és:

```
1 options { directory "/var/named"; }
```

Per cada fitxer de zona caldrà definir una entrada al fitxer de configuració de BIND indicant el nom de la zona, el tipus i el nom del fitxer.

La sintaxi seria:

```
1 zone "nom_zona" in { type master|slave|hint; file "nom_fitxer_zona";
```

Un exemple de configuració de zona podria ser aquest:

```
1 zone inf.ioc.cat in { type master; file "db.inf.ioc.cat; }
```

La descripció de cada camp és la següent:

- **zone nomZona**: Com es pot veure en l'exemple es defineix una zona corresponent al domini *inf.ioc.cat*.
- **type master | slave | hint**: El servidor serà primari (*master*) i autoritatiu per a aquesta zona. El camp tipus pot prendre els valors *master*, *slave* i *hint* que signifiquen el següent:
  - *master*: El servidor és amb autoritat per a aquesta zona i gestiona els fitxers de zona.
  - *slave*: El servidor és autoritari per a la zona però obté les dades de la zona del primari o *master*.
  - *hint*: Indica que es tracta de la informació corresponent als servidors de noms de la zona arrel. Aquesta informació té un tractament especial diferent del de les altres zones.
- **file nom_fitxer_zona**: Indica el fitxer amb els registres de recurs de la zona. En l'exemple és el fitxer *db.inf.ioc.cat*.

Cal posar especial atenció en la definició dels fitxers de zona per a la resolució inversa, utilitzant per exemple la xarxa 192.168.20.0/24. La zona s'anomena *20.168.192.in-addr.arpa* i el fitxer de zona *db.192.168.20*. El nom del fitxer conté l'adreça de xarxa en l'ordre natural, però el domini té els octets invertits perquè forma part del domini *in-addr.arpa*.

2.6.2 El fitxer de configuració DNS

Configuració bàsica

Per fer funcionar el servidor DNS cal configurar-lo prèviament. Per poder arrancar li cal saber quin és el domini que administrarà i quins són els noms de màquina que pertanyen al domini (definir els registres de recurs a utilitzar) entre molts altres paràmetres de configuració.

El paquet dns conté un fitxer d'exemple de configuració d'una zona al directori */user/share/doc/bind*/sample/etc/named.conf*. Aquest fitxer es pot copiar a */etc/-named.conf* i passarà a ser la configuració bàsica del servidor DNS.

```
[root@portatil ~]# rpm -ql bind | grep named.conf
/etc/dbus-1/system.d/named.conf
/etc/named.conf
/usr/share/doc/bind-9.4.2/sample/etc/named.conf
/usr/share/man/man5/named.conf.5.gz
```

Podeu fer la llista del contingut fent:

```
[root@portatil ~]# ll /usr/share/doc/bind-9.4.2/sample/etc/named.conf
-rw-r--r-- 1 root root 4273 14 jun  2006 /usr/share/doc/bind-9.4.2/sample/etc/
    named.conf
```

Observem-ne el contingut:

```
[root@portatil ~]# cat /usr/share/doc/bind-9.4.2/sample/etc/named.conf
//
// Sample named.conf BIND DNS server 'named' configuration file
// for the Red Hat BIND distribution.
//
// See the BIND Administrator's Reference Manual (ARM) for details, in:
//   file:///usr/share/doc/bind-*/arm/Bv9ARM.html
// Also see the BIND Configuration GUI : /usr/bin/system-config-bind and
// its manual.
//
options
{
        /* make named use port 53 for the source of all queries, to allow
         * firewalls to block all ports except 53:
         */
        query-source    port 53;
        query-source-v6 port 53;

        // Put files that named is allowed to write in the data/ directory:
        directory "/var/named"; // the default
        dump-file               "data/cache_dump.db";
        statistics-file         "data/named_stats.txt";
        memstatistics-file      "data/named_mem_stats.txt";

};
logging
{
/*      If you want to enable debugging, eg. using the 'rndc trace' command,
 *      named will try to write the 'named.run' file in the $directory (/var/named).
 *      By default, SELinux policy does not allow named to modify the /var/named directory,
 *      so put the default debug log file in data/ :
 */
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
//
// All BIND 9 zones are in a "view", which allow different zones to be served
// to different types of client addresses, and for options to be set for groups
// of zones.
//
// By default, if named.conf contains no "view" clauses, all zones are in the
// "default" view, which matches all clients.
//
// If named.conf contains any "view" clause, then all zones MUST be in a view;
// so it is recommended to start off using views to avoid having to restructure
// your configuration files in the future.
//
view "localhost_resolver"
{
/* This view sets up named to be a localhost resolver ( caching only nameserver ).
 * If all you want is a caching-only nameserver, then you need only define this view:
 */
        match-clients           { localhost; };
        match-destinations      { localhost; };
        recursion yes;
        # all views must contain the root hints zone:
        include "/etc/named.root.hints";

        /* these are zones that contain definitions for all the localhost
         * names and addresses, as recommended in RFC1912 - these names should
         * ONLY be served to localhost clients:
         */
```

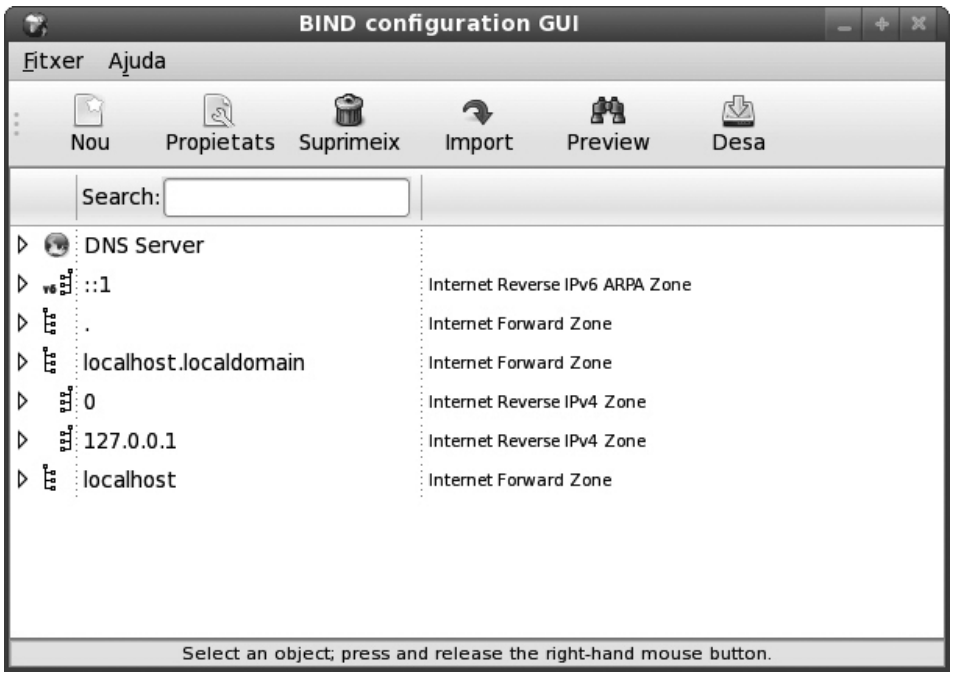
```
65          include "/etc/named.rfc1912.zones";
66  };
67  view "internal"
68  /* This view will contain zones you want to serve only to "internal" clients
69     that connect via your directly attached LAN interfaces - "localnets" .
70   */
71          match-clients           { localnets; };
72          match-destinations      { localnets; };
73          recursion yes;
74          // all views must contain the root hints zone:
75          include "/etc/named.root.hints";
76
77          // include "named.rfc1912.zones";
78          // you should not serve your rfc1912 names to non-localhost clients.
79
80          // These are your "authoritative" internal zones, and would probably
81          // also be included in the "localhost_resolver" view above :
82
83          zone "my.internal.zone" {
84                  type master;
85                  file "my.internal.zone.db";
86          };
87          zone "my.slave.internal.zone" {
88                  type slave;
89                  file "slaves/my.slave.internal.zone.db";
90                  masters { /* put master nameserver IPs here */ 127.0.0.1; } ;
91                  // put slave zones in the slaves/ directory so named can update
                        them
92          };
93          zone "my.ddns.internal.zone" {
94                  type master;
95                  allow-update { key ddns_key; };
96                  file "slaves/my.ddns.internal.zone.db";
97                  // put dynamically updateable zones in the slaves/ directory so
                        named can
98  update them
99          };
100 };
101 key ddns_key
102 {
103         algorithm hmac-md5;
104         secret "use /usr/sbin/dns-keygen to generate TSIG keys";
105 };
106 view    "external"
107 {
108 /* This view will contain zones you want to serve only to "external" clients
109  * that have addresses that are not on your directly attached LAN interface
        subnets:
110  */
111         match-clients           { !localnets; !localhost; };
112         match-destinations      { !localnets; !localhost; };
113
114         recursion no;
115         // you'd probably want to deny recursion to external clients, so you
              don't
116         // end up providing free DNS service to all takers
117
118         // all views must contain the root hints zone:
119         include "/etc/named.root.hints";
120
121         // These are your "authoritative" external zones, and would probably
122         // contain entries for just your web and mail servers:
123
124         zone "my.external.zone" {
125                 type master;
126                 file "my.external.zone.db";
127         };
128 };
```

En la configuració per defecte es poden analitzar els diversos elements que es configuren:

- **options**: En la secció options s'hi defineixen les opcions genèriques del servidor DNS.
- **logging**: es defineix com serà el procés d'enregistrament dels logs del servei.
- **localhost_resolver**: aquesta secció permet definir el servidor DNS com un servidor només *cache*. És a dir, no és autoritari de cap domini, no gestiona cap domini, cap zona, no té fitxers de zona, l'única funció que fa és de servidor DNS *cache*.
- **internal**: aquesta secció permet definir les zones i zones delegades que es volen gestionar amb el servidor. Es donarà servei a les xarxes locals internes que es defineixen en aquesta secció.
- **external**: defineix el servei a oferir a clients externs a la xarxa local. És per oferir serveis DNS a clients exteriors.

La figura 2.3 mostra una eina gràfica que permet la configuració del servei de noms.

FIGURA 2.3. Applet gràfic de gestió del servidor bind



En sistemes GNU/Linux és usual que els servidors es gestionin editant directament els fitxers de configuració. Es disposa, però, d'applets gràfics que permeten fer aquesta mateixa tasca des de l'entorn gràfic. Usualment no fan res propi, és a dir, simplement serveixen per modificar els fitxers de text de configuració a través d'un entorn gràfic més amable.

Exemple de configuració bàsica

Com heu pogut veure el fitxer d'exemple que proporciona *bind* és força complex. Una configuració per donar servei a una escola podria ser la següent:

Fitxer de configuració del servei named que es troba a */etc/named.conf*:

```
// named.conf for inf.escola.org
options {             // definició d'opcions globals
        directory "/var/named";
        // query-source port 53;
   forward only;
   forwarders {
  100.1.1.201;
  100.1.1.202;
 };
};

//zone "." {            // definició dels nodes arrel
//        type hint;
//        file "root.hints";
//};

//zone "0.0.127.in-addr.arpa" {     // definició del localhost
//        type master;
//        file "localhost.rev.zone";
//};

zone "inf.escola.org" {        // definició per a la resolució directa
        notify no;
        type master;
        file "inf.escola.org.zone";
};

zone "0.16.172.in-addr.arpa" {     // definició per a la resolució inversa
        notify no;
        type master;
        file "inf.escola.org.rev.zone";
};
```

El fitxer de zona per a la resolució directa anomenat en l'exemple anterior*inf.escola.org.zone* conté els registres de recurs que defineixen la zona:

```
; Zone file for inf.escola.org
$TTL 3D
@ INSOA server.inf.escola.org. postmaster.inf.escola.org. (
                                2007101910; Serial
                                8H ; Refresh
                                2H ; Retry
                                4W ; Expire
                                1D); Minimum TTL
 NS      server
 MX      10 mailhost
 A  172.16.0.10

; loopback
localhostA       127.0.0.1

; Ordinadors del departament
; subxarxa 172.16.0.0 mask 255.255.255.192
server     A172.16.0.10
mailhost   A172.16.0.10
www        CNAME   server
ftp        CNAME   server
ldap       CNAME   server

; Servidors, router, i impressora
router     A 172.16.0.1
hp-7200c   A 172.16.0.5

; Estacions  departament. Adreça IP fixa i automàtica des de DHCP.
pcprofe01 A 172.16.0.15
pcprofe02 A 172.16.0.16
```

```
31
32 ; Estacions aula-1. Adreca IP fixa i automàtica des de DHCP.
33 ; subxarxa   172.16.0.128 mask 255.255.255.192
34 switch-AA 172.16.0.251
35 pc01 A 172.16.0.131
36 pc02 A 172.16.0.132
```

Per a cada xarxa client que es defineix cal definir el fitxer de resolució inversa de la xarxa en el domini *in-addr.arpa*. El fitxer de resolució inversa correspon al nom *inf.escola.org.rev.zone*. Aquí hi ha una entrada PTR per a cada host definit en la resolució directa.

```
1  ; Zone file for 0.168.192.in-addr.arpa
2  $TTL 3D
3  @INSOA server.inf.escola.org. postmaster.inf.escola.org. (
4      2007101910; Serial
5      8H      ; Refresh
6      2H      ; Retry
7      4W      ; Expire
8      1D)     ; Minimum TTL
9   NS  server.inf.escola.org.
10 ; departament
11 1   IN PTR router.inf.escola.org.
12 5   IN PTR hp-7200c.inf.escola.org.
13 10  IN PTR server.inf.escola.org.
14 15  IN PTR pcprofe01.inf.escola.org.
15 16  IN PTR pcprofe02.inf.escola.org.
16 131 IN PTR pc01.inf.escola.org.
17 132 IN PTR pc02.inf.escola.org.
18 251 IN PTR switch-A.inf.escola.org.
```

2.6.3 El client DNS: resolver, hosts i nsswitch

Un equip client que vol resoldre un nom de *host* té diferents maneres de fer-ho. Es pot fer localment mitjançant un fitxer de *hosts* (típicament el /etc/hosts) o distribuïdament usant DNS (el *resolver*). De fet, es poden aplicar tots dos mètodes conjuntament indicant-ne la precedència en algun fitxer de configuració del sistema (en sistemes Unix, el fitxer /etc/ nsswitch.conf).

Resolver

El *resolver* és la part client del sistema de noms de dominis DNS, que està organitzat en una estructura client-servidor. Cada *resolver* implementa les seves opcions, però n'hi ha que són suficientment genèriques per descriure-les. En la majoria de sistemes Unix, aquestes opcions es defineixen en el fitxer /etc/resolv.conf.

Les següents són les directives del fitxer /etc/resolv.conf:

- **Domain** (*local domain name* o nom de domini local) indica el nom de domini del *host* al qual pertany el *resolver*, serveix per completar els noms de domini que no són qualificats (FQDN). Per exemple, amb el valor de domain *inf.ioc.cat.*, si es vol resoldre *pc30* (un nom de *host* no

qualificat) se li afegirà el nom de domini indicat; per tant, s'intentarà resoldre *pc30.inf.ioc.cat.* Generalment, la directiva *domain* és excloent de la directiva *search*.

- **Search** permet modificar el comportament per defecte indicant explícitament la llista de dominis a aplicar. El primer d'aquests és aplicat com el nom del domini local (*local domain name*) i és per això que la directiva *search* és excloent de la directiva *domain*. Si per exemple s'utilitza *search inf.ioc.cat inf.ioc.net gencat.cat* per resoldre el nom de *host pc30*, s'aplicarà cada domini seqüencialment*pc30.inf.ioc.cat*, *pc30.inf.ioc.net* i *pc30.gencat.cat*.

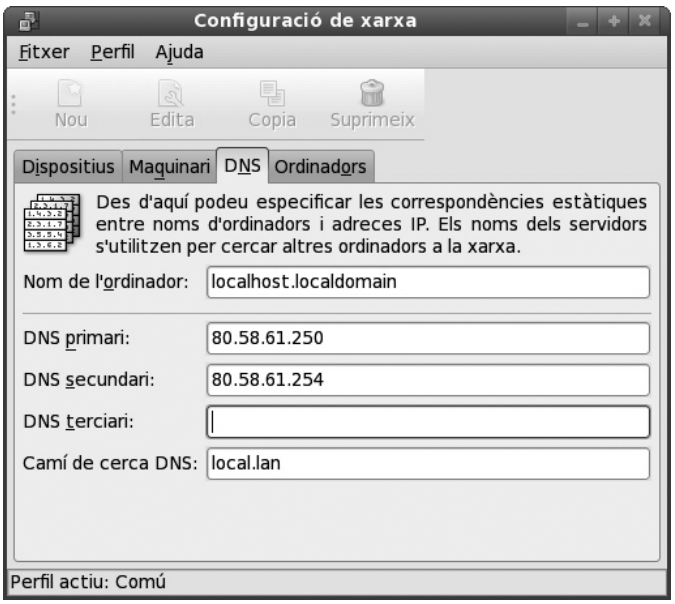
- **Nameserver** permet especificar el servidor de noms a utilitzar. Se'n poden indicar fins a tres per si no hi ha accés al servidor. El *resolver* intenta connectar amb el primer servidor i si ho aconsegueix realitza les consultes a aquest servidor. Si el servidor no és accessible intenta el següent, i així també per a l'últim. Fixeu-vos que disposar de tres servidors de noms no significa que si el primer no pot respondre la consulta es repeteix la mateixa consulta al següent. En general, els servidors de noms no restringeixen l'accés de manera que qualsevol *host* hi pot fer consultes.

Criteri de resolució

Per defecte, quan cal resoldre un nom de *host* (i no s'ha especificat la directiva *search*), el *resolver* fa el següent: si el nom de *host* inclou un punt (*pc30.inf*) mira de resoldre'l tal qual i, si no pot, hi aplica el nom de domini (*pc30.inf.inf.ioc.cat.*). Si el nom de *host* no conté cap punt (*pc30*), primer s'afegeix el domini i es mira de resoldre (*pc30.inf.ioc.cat.*), i si no es troba es mira de resoldre tal qual (*pc30*).

La figura 2.4 mostra una interfície gràfica que permet configurar els servidors DNS.

Figura 2.4. Applet gràfic de configuració del resolver



En aquest exemple s'observa que s'han definit dos servidors de noms DNS,un de primari i un de secundari. La directiva search correspon al valor local.lan. Tal com és tradicional en sistemes GNU/Linux i Unix, aquest applet gràfic simplement modifica el fitxer de configuració /etc/resolv.conf.

Servidor de noms d'una altra organització

Es poden configurar els *hosts* perquè utilitzin el servidor de noms d'una altra organització (perquè és més ràpid, per estalviar-se feina, etc.), però no és una bona pràctica.

```
# Exemple de fitxer de configuració client /etc/resolv.conf
[root@portatil ~]# cat /etc/resolv.conf
; generated by /sbin/dhclient-script
search ioc.cat
nameserver 80.58.61.250
nameserver 80.58.61.254
```

nsswitch

Un *host* que vol fer la resolució d'un nom pot optar per fer-la localment amb el fitxer *hosts* o usant DNS amb el *resolver*. De fet, els pot utilitzar tots dos, simplement ha d'indicar en quin ordre. En els sistemes Unix hi ha el fitxer de configuració /etc/nsswitch.conf, que entre altres coses permet configurar l'ordre de les resolucions de noms.

Una entrada tipus **hosts: dns files** indica que primer es mira de resoldre el nom per DNS i si no es pot es busca al fitxer de *hosts* local. Una entrada tipus **hosts: files dns** indica que primerament es buscarà en el fitxer de*hosts* local i posteriorment el DNS. Aquesta opció acostuma a ser més utilitzada si es permet que un *host* pugui personalitzar els noms del seu entorn i tinguin precedència sobre els noms del DNS.

```
# Exemple de fitxer de selecció de tipus de resolució:
[root@portatil ~]# cat /etc/nsswitch.conf
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#       nisplus or nis+         Use NIS+ (NIS version 3)
#       nis or yp               Use NIS (NIS version 2), also called YP
#       dns                     Use DNS (Domain Name Service)
#       files                   Use the local files
#       db                      Use the local database (.db) files
#       compat                  Use NIS on compat mode
#       hesiod                  Use Hesiod for user lookups
#       [NOTFOUND=return]       Stop searching if not found so far
#

# To use db, put the "db" in front of "files" for entries you want to be
# looked up first in the databases
#
# Example:
#passwd:    db files nisplus nis
#shadow:    db files nisplus nis
#group:     db files nisplus nis

passwd:     files
shadow:     files
group:      files

#hosts:     db files nisplus nis dns
hosts:      files dns
```

Fitxer de hosts

En tot sistema sempre es poden personalitzar els noms dels *hosts* mitjançant un fitxer local, generalment anomenat *hosts* (/etc/hosts en sistemes Unix). És un fitxer de text net que conté una entrada per línia del tipus **ip nom_canònic alias1 alias2 ... aliasn**. Per a una IP determinada es defineix el seu nom canònic (nom únic

identificatiu) i altres noms que actuen com a àlies del nom canònic. Per exemple, 192.168.0.1 server.inf.ioc.cat server machine defineix els àlies *server* i *machine* per al *host server.ioc.cat*.

El fitxer de *host* es pot utilitzar en un entorn de xarxa petit, però no es pot escalar a grans xarxes. Caldria posar dins el fitxer els noms de tots els*hosts* de la xarxa i copiar el fitxer en cada *host*. Justament per evitar aquesta dificultat, es va desenvolupar el sistema DNS.

```
# Fitxer d'exemple lmhosts.sam d'un sistema operatiu Windows:
# Copyright (c) 1993–1999 Microsoft Corp.
#
# Este es un archivo LMHOSTS de ejemplo utilizado por TCP/IP de Microsoft
# para Windows.
#
# Este archivo contiene las direcciones de IP asociadas con nombres (NetBIOS)
# de equipos. Cada entrada debería encontrarse en una línea individual. La
# dirección de IP debería colocarse en la primera columna seguida del nombre de
# equipo correspondiente. La dirección de IP y el nombre de equipo
# deberían estar separados como mínimo por un espacio o tabulador. El carácter
# "#" se usa generalmente para indicar el principio de un comentario (consulte
# las excepciones ms adelante).
#
# Este archivo es compatible con los archivos lmhosts de TCP/IP de Microsoft
# LAN Manager 2.x y ofrece las siguientes extensiones:
#
#      #PRE
#      #DOM:<dominio>
#      #INCLUDE <nombre_de_archivo>
#      #BEGIN_ALTERNATE
#      #END_ALTERNATE
#      \0xnn (permite caracteres no imprimibles)
#
# Cualquier entrada iniciada con "#PRE" provocar  que la entrada se cargue
# en la ‡cach. De forma predeterminada, las entradas no son cargadas,
# sino que se consultan sólo cuando la resolución dinámica de nombres falla.
#
# Cualquier entrada iniciada con "#DOM:<dominio>" asociar la entrada
# con el dominio especificado en <dominio>. Esto influye en cómo los servicios
# Examinador e Inicio de sesión actúa en entornos TCP/IP. Para cargar una
    entrada
# #DOM es necesario agregar #PRE a la línea.
#
# Especificando "#INCLUDE <nombre_de_archivo>" obligar al software RFC
# NetBIOS (NBT) a buscar y tratar el <nombre_de_archivo> como si fuera
# local. <nombre_de_archivo> es generalmente un nombre basado en UNC,
# permitiendo que un lmhosts central sea mantenido en un servidor. Si el
    servidor
# se encuentra fuera del área de difusión, probablemente será necesario
# proporcionar al servidor información sobre direcciones de servidores antes
# de #INCLUDE.
#
# #BEGIN_ y #END_ALTERNATE permiten agrupar múltiples sentencias
# #INCLUDE. Cualquier éxito individual se considera éxito del grupo.
#
# Finalmente, caracteres no imprimibles pueden ser incluidos en este tipo de
# archivos, primero se ponen entre comillas los nombre NetBIOS y después
# se utiliza la notación \0xnn para especificar un valor hexadecimal para un
# carácter no imprimible.
#
# El siguiente ejemplo ilustra todas estas extensiones:
#
# 102.54.94.97     rhino               #PRE   #DOM:conectividad  #red DC de
    grupo
# 102.54.94.102    "nombreap  \0x14"         #servidor ap especial
# 102.54.94.123    popular            #PRE   #servidor origen
# 102.54.94.117    localsrv           #PRE   #necesario para "#INCLUDE"
# #BEGIN_ALTERNATE
# #INCLUDE \\localsrv\public\lmhosts
```

```
58 # #INCLUDE \\rhino\public\lmhosts
59 # #END_ALTERNATE
60 #
61 # En el ejemplo anterior, el servidor "nombreap" contiene un carácter
62 # especial en su nombre, el nombre de servidor "popular" es cargado, y
63 # el nombre de servidor "rhino" es especificado, por lo que puede ser
64 # utilizado para un posterior #INCLUDE de archivo lmhosts central si
65 # el sistema "localsrv" no está disponible.
66 #
67 # Tenga en cuenta que en cada búsqueda se analiza el archivo completo,
       incluyendo los
68 # comentarios, por lo que mantener la mínima cantidad de comentarios mejora el
       rendimiento.
69 # Sin embargo no es siempre recomendable agregar sólo las entradas de archivo
       LMHOSTS al final
70 # de este archivo.
```

La figura 2.5 mostra una interfície gràfica de gestió de xarxa des d'on configurar noms de *host* i àlies.

FIGURA 2.5. Applet gràfic per configurar noms de host

Així per exemple podem observar que el pc01 corresponent a l'adreça IP 192.168.1.32 també es pot anomenar desk1, pere i usuari1.

2.7 Realització de transferències entre dos o més servidors

Els dominis d'Internet s'administren en zones, cada una gestionada per dos o més servidors de noms. Segons l'estàndard, cada zona té un servidor primari i almenys un o més servidors secundaris. Ambdós són autoritaris per a la zona que administren. Caldrà, doncs, que aquests servidors disposin d'informació tan coherent com sigui possible, que comparteixin la mateixa informació. Això es realitza mitjançant les transferències de zona.

Sovint els servidors de noms actuen també com a *cache*, emmagatzemant en la memòria les respostes d'altres servidors per tal d'incrementar l'eficiència. Quan emeten aquestes respostes actuen de forma no autoritària.

Primari/secundari(s): una zona és gestionada per un servidor primari i un o més servidors secundaris o de *backup*.
Autoritari/no autoritari: els servidors d'una zona (el primari i els secundaris) són autoritat per a aquella zona que administren. Les respostes que emeten basant-se en la informació del *cache* (i no de la base de dades de zona) són no autoritàries.
Transferència de zona: la informació de la base de dades de zona ha de ser coherent entre els servidors primari i secundaris. La transferència de zona és el mecanisme que s'estableix per fer que comparteixin la mateixa informació.

2.7.1 Autoritari, no autoritari i informació de base de dades de zona

Cada zona de l'espai dels noms de domini és gestionada per un o més servidors autoritaris per a la zona. Significa que són els servidors que manen, que tenen l'última paraula respecte de la zona. De fet significa que són els servidors que administren la zona. Aquests servidors es configurem com a **autoritaris** de la zona. Les respostes que emeten a consultes referents a la seva zona porten un segell indicant que són autoritàries, que provenen de la font de la informació de la base de dades distribuïda de l'espai de noms de domini.

Els servidors de noms guarden en un emmagatzematge *cache* les respostes que reben d'altres servidors. Aquesta informació també és utilitzada per elaborar respostes a consultes de dominis fora de la zona de la qual són autoritaris. És a dir, quan un servidor rep d'un altre la llista de servidors autoritaris (en una consulta interactiva) per una zona, aquesta llista s'emmagatzema al *cache*. Quan un servidor rep una resposta a una consulta també es guarda. Quan es rep una resposta negativa, indicant que el domini de la consulta o el tipus de dada sol·licitat no existeix, també es guarda. S'anomena negative *caching*.

Un servidor respon utilitzant la informació de la base de dades de la seva zona, però també respon a altres consultes utilitzant la informació existent en el seu *cache*. Quan es repeteix una consulta efectuada abans, s'utilitza la resposta del *cache*. També s'utilitza del *cache* la llista de servidors autoritaris per una zona més propera al domini que es busca, per tal de no preguntar als servidors arrel. Quan un servidor de noms respon utilitzant la informació del *cache*, la resposta és no autoritativa. No prové del servidor autoritari de la zona sinó que s'utilitza una informació prèviament obtinguda (i que pot estar desfasada).

Per exemple, primerament s'ha fet una consulta per al domini *adm.ioc.cat*, i en el procés de resolució recursiu el servidor ha obtingut la llista de servidors autoritaris per als dominis .cat i *ioc.cat.* Si ara es demana al mateix servidor pel domini *inf.ioc.cat* primerament provarà de resoldre aquest domini, però com que és desconegut el següent domini més pròxim és *ioc.cat* i aquest sí que el coneix, perquè el té emmagatzemat al *cache* de la resposta anterior. S'utilitzarà la llista de servidors autoritaris del domini *ioc.cat* per obtenir una resposta o per continuar descendint per l'arbre de l'espai de noms.

Emmagatzemar informació de les respostes d'altres servidors en la *cache* proporciona dos grans avantatges:

1. Incrementa la velocitat de resposta. Ja no cal anar a trobar la resposta a la font de dades de la zona, sinó que s'utilitza la informació d'una resposta anterior.
2. Evita la sobrecàrrega dels servidors arrel.

No cal anar per a cada consulta al node arrel un cop es disposa al *cache* d'informació més propera al domini a cercar. La utilització del *cache* té, però, un inconvenient que cal mesurar bé, les dades que s'utilitzen no necessàriament són actualitzades ni reflecteixen l'estat actual de la xarxa. DNS es basa en una base de dades jeràrquica i distribuïda on la informació es troba en els fitxers gestionats per cada servidor de zona. Una dada emmagatzemada en el *cache* pot no reflectir la dada real que ha estat modificada en el servidor autoritari de zona, però que encara no s'ha propagat perquè la dada es manté en el *cache* fins a caducar el TTL (temps de vida de la dada).

Són servidors **autoritaris** els que administren una zona (tant el servidor primari com els secundaris). Tenen accés a la informació de la base de dades de zona.
Són respostes **no autoritàries** les que provenen de la informació desada en la memòria *cache*.

Fixeu-vos que si la informació del *cache* es desa indefinidament, els canvis que es fessin en els servidors autoritaris no es propagarien als altres servidors (perquè segurament ja disposarien d'una resposta en el *cache* anterior). Cal un mecanisme perquè la informació del *cache* caduqui transcorregut un cert interval de temps.

Anomenem **TTL** time to live **(temps de vida)** l'interval de temps que les dades han de perdurar en el *cache*; un cop transcorregut, les dades s'eliminen. Si fa falta una dada, per obtenir-la caldrà fer una consulta de nou. Si el TTL és un interval de temps petit es facilita la propagació de les actualitzacions (millor consistència de la informació), però es penalitza el rendiment, ja que caldrà fer consultes més sovint (tornar a les fonts de dades). Si el TTL és un interval de temps gran el rendiment millora (més consultes són contestades directament del *cache* en lloc de consultar la font de dades), però fa que la propagació dels canvis sigui més lenta. Això provoca menys consistència de la base de dades distribuïda de noms i incrementa la possibilitat de proporcionar informació errònia.

El temps del TTL l'estableix l'administrador de sistemes. Ha de saber buscar un compromís entre el rendiment (TTL més gran implica menys trànsit de consultes i consultes més ràpides) i la consistència de la informació (TTL més petit significa propagació més fàcil).

L'ús de l'emmagatzematge de respostes en memòria *cache* no és exclusiu del servidor de noms. Sovint el *resolver* també implementa una petita*cache* per proporcionar més rendiment i velocitat de resposta a les aplicacions. Fins i tot

algunes aplicacions (per exemple el navegador web Firefox) implementen la seva pròpia *cache* per millorar el rendiment.

2.7.2 Servidor primari/secundari

L'estàndard que defineix el DNS (*domain name system*) estableix que cal configurar dos o més servidors autoritaris per a cada zona anomenats servidor primari i servidor secundari. El motiu és proporcionar un mecanisme de redundància, robustesa, rendiment i *bakcup*. Si el servidor de noms falla i és únic possiblement la xarxa caurà, serà inoperativa. Per una banda els usuaris no saben les adreces IP dels *hosts*, dels serveis, de les seus web, etc. als quals volen accedir. D'altra banda, molts serveis de xarxa requereixen la resolució DNS per funcionar. Per tant, el servidor de noms és un punt crític en tota xarxa. Un segon servidor autoritari proporciona redundància i robustesa en cas que el primari caigui. Proporciona més rendiment perquè les consultes es poden repartir, balancejar, entre els dos o més servidors. I proporciona un mecanisme de *backup* o salvaguarda amb la repetició de les dades.

L'estàndard defineix que calen almenys dos servidors i segur que estem avesats a configurar clients de xarxa omplint els camps del DNS primari i DNS secundari del proveïdor d'Internet a casa. De totes maneres, ningú obliga a seguir els estàndards. En una xarxa petita amb pocs equips i pocs requeriments podem utilitzar un únic servidor DNS. Sabem, però, que si cau no funcionarà la xarxa.

El servidor primari és el que manté la base de dades de la zona i el que l'actualitza. El servidor secundari obté una còpia de la base de dades de la zona a través del primari. Existeix un mecanisme d'actualització d'informació de zona entre el servidor primari i el secundari.

El servidor **primari** manté la base de dades de la zona i l'actualitza. Aquesta informació es copia als servidors **secundaris** utilitzant un procés de **transferència**.
Establir com i quan s'ha de fer aquesta transferència és important per proporcionar un bon servei DNS. Cal buscar un **compromís** entre **actualitzar** constantment la informació o disposar d'informació **caducada**.

Es poden definir més servidors secundaris per millorar el rendiment del servei de resolució de noms. Si hi ha moltes consultes o es té molta por a una caiguda del sistema es poden configurar més servidors, així les consultes es poden repartir entre tots ells. De totes maneres el més usual és un servidor primari i un de secundari. Recordeu que tant el servidor primari com el secundari són autoritat, tot i que només el primari té els fitxers de zona. El secundari n'obté una còpia.

Es pot configurar un servidor de noms de domini perquè actuï fent la funció de només *cache*. No gestiona cap zona, simplement atén les peticions dels *resolver* client i les passa a altres servidors de noms. La seva funció és emmagatzemar en

cache les respostes que obté abans de passar-les als clients. Això li permetrà que les futures consultes que siguin repetides les pugui contestar directament en lloc de demanar-les a un altre servidor. Evidentment aquest tipus de servidor no emet respostes amb autoritat.

El següent exemple mostra la llista dels servidors de noms de *Google* i de l'*IOC*:

```
[root@localhost ~]# host -t NS google.com
google.com name server ns2.google.com.
google.com name server ns4.google.com.
google.com name server ns3.google.com.
google.com name server ns1.google.com.

[root@localhost ~]# host -t NS ioc.cat
ioc.cat name server dns2.nominalia.com.
ioc.cat name server dns1.nominalia.com.
```

2.7.3 Transferència de zones

El protocol DNS indica que cal utilitzar almenys dos servidors DNS per a una zona, un serà el *master* o primari i els altres *slaves* o secundaris. Tots aquests seran autoritat per a la zona. L'administrador fa el manteniment únicament dels fitxers de zona del servidor primari (conté els originals). Els servidors secundaris obtenen els seus fitxers de configuració de zona del primari o d'un altre de secundari. El procés de transferència de la informació de la zona entre primari i secundari és automatitzat i és transparent per a l'administrador.

Els fitxers de zona no fan cap referència al fet que el nameserver que els utilitza sigui primari o secundari, és en el fitxer de configuració global del servei DNS (el fitxer *named.conf* per a l'aplicació BIND 9) on cal indicar-ho. De fet, els fitxers de zona es transfereixen del primari als secundaris i han de contenir la mateixa informació.

Generalment els servidors secundaris disposen d'una còpia local pròpia del fitxer de *root* i del de *loopback*, ja que sempre és el mateix fitxer. Les altres configuracions ens transfereixen des del primari. Per tant, un servidor DNS secundari és primari per al domini *root* i *loopback* i secundari per a la resta de la zona.

L'administrador del sistema en posar en funcionament un servidor secundari ha de copiar-hi els fitxers corresponents al *loopback*, a la zona *root* (*db.cache*) i el fitxer de configuració global *(/etc/named.conf* en BIND 9). En el fitxer de configuració global caldrà modificar l'atribut *type* amb el valor *slave*; l'atribut *file* amb un nom tipus *bak.db.<nom_zona>* i afegir un atribut *masters* amb la llista de les adreces IP que són autoritaris de la zona. Els canvis caldrà fer-los en totes les definicions de zones excepte per al *loopback* i per a *root hints*.

```
# extracte d'un fitxer /etc/named.conf
zone "example.com" {     // definició de la zona on és autoritari
    type master;
    file "example.com.db";
    allow-transfer {     // Ips dels servidors secundaris als quals es permet
        192.168.4.14;    // transferir la zona
```

```
        192.168.5.53; };
};
zone "." {          // definició dels nodes arrel
        type hint;
        file "root.hints"; };
zone "0.0.127.in-addr.arpa" {  // definició del localhost
        type master;
        file "localhost.rev.zone"; };
```

En l'exemple anterior es defineixen els fitxers per a la resolució inversa del *loopback* i el fitxer de resolució dels nodes arrel, que sempre es fa de la mateixa manera. La part interessant és on es defineix el mateix servidor com a *màster* de la zona *example.com* i els servidors *192.168.4.14* i *192.168.5.53* com a secundaris d'aquesta zona.

Es poden posar diverses adreces IP amb servidors autoritzats de la zona en l'atribut *masters*. Segons sigui el programa/versió que s'utilitzi de servidor DNS s'aplicarà un criteri d'elecció. Els fitxers que es transfereixen del servidor primari al secundari no cal desar-los localment en el servidor secundari (la informació resideix a memòria), però s'aconsella fer-ho utilitzant l'atribut *file*. D'aquesta manera el servidor secundari pot continuar treballant, encara que el servidor primari no sigui accessible (fins que li caduquin les dades).

```
# fragment d'un fitxer /etc/named.conf d'un servidor secundari
zone "eng.example.com" {
    type slave;
    file "eng.example.com.bk";
    masters { 192.168.4.12; 192.168.4.13};     // Ips dels servidors primàris
};
```

En l'exemple anterior s'observa que aquest servidor es defineix amb l'atribut *slave* per a la zona *eng.example.com*. També s'indiquen quins són els servidors primaris d'aquesta zona, d'on pot rebre la transferència de zona. Usualment el primer és el primari i si n'hi ha més d'un són altres secundaris.

L'actualització de la informació entre el servidor primari i els secundaris és molt important. Establir correctament aquests paràmetres és un art. Un servidor amb informació antiga causa un mal funcionament a la xarxa, un servidor actualitzat constantment consumeix els recursos de la xarxa.

Per cada fitxer de zona hi ha una entrada SOA que conté els valors necessaris per governar les transferències. Per exemple en l'entrada:

```
inf.ioc.cat. IN SOA ns1.inf.ioc.cat. admin.ns1.inf.ioc.cat. ( 1h 3h 1h 1w 1h )
```

Els camps entre parèntesis corresponen a:

- *serial (1h)*: valor que indica el número de versió de les dades. A cada actualització s'incrementa. Quan el secundari contacta al primari mira aquest valor, si en el màster és superior sol·licita actualitzar la zona.
- *refresh (3h)*: és l'interval de temps cada quan el secundari ha de contactar amb el primari per comprovar si cal actualitzar la zona.

- *retry (1h)*: si no pot contactar amb el primari en fer un *refresh* ho torna a intentar passat aquest interval de temps.
- *expire (1w)*: si el primari no és accessible aquest és el temps que el secundari pot continuar responent a les consultes de la zona. Un cop transcorregut ha de deixar de fer-ho, perquè es considera que les dades són massa antigues i desfasades.
- *negative caching ttl (1h)*: indica el temps que ha de mantenir a la*cache* les respostes negatives el secundari (dominis no trobats).

2.8 Comprovació de funcionament del servei

En informàtica (a diferència del que passa en l'aviació, per exemple), normalment es comprova si funcionen les coses provant-les. Si el servei DNS emet les respostes adequades, és que funciona. Possiblement la manera mes fàcil de fer-ho és fent ping als noms de domini que es volen comprovar.

Existeixen moltes eines per fer consultes DNS simulant el funcionament d'un client i que permeten observar la resposta del servidor. Les més populars són les utilitats nslookup, dig i host. Aquestes ordres permeten obtenir informació dels recursos d'un servidor DNS, resoldre consultes i saber quins són els seus registres de recurs.

Altres eines que cal saber utilitzar són la utilitat nmap, que permet analitzar els ports d'un *host* i *sniffers* de xarxa que permetin monitorar tot el trànsit DNS que s'està produint. Les utilitats iptraf i wireshark són perfectes per observar el trànsit TCP i UDP que pot generar el protocol DNS.

2.8.1 Comprovació del funcionament bàsic

Comprovació DNS

L'administrador pot verificar el funcionament del servidor DNS observant:

- l'estat del servei (*on*, *off*)
- el PID del servei (ha d'estar *running*).
- el registre de *logs*.
- monitorar el trànsit de xarxa amb una eina tipus wireshark.
- l'estat dels ports.

Comprovar que el servidor DNS està en funcionament és un procés ben senzill, n'hi ha prou de comprovar que el servei està engegat. Això no vol dir, en cap cas, que el servei estigui funcionant correctament. Potser el servidor està engegat però no està correctament configurat. De fet, la configuració és la part realment important de l'administració d'un servei i també del servei DNS.

A part de comprovar l'estat del servei (amb l'opció *status*), l'administrador pot assegurar-se que el dimoni del servei està en execució buscant el seu PID (*process identifier* o indicador de número de procés). Una altra activitat a fer és monitorar el registre d'activitats del servei (els *logs*). Tot el trànsit DNS és trànsit de xarxa TCP/IP; per tant, també es pot observar l'estat dels ports i analitzar el trànsit que s'hi produeix.

Tot procés en el sistema té un identificador de procés. El PID dels serveis usualment es desen en el sistema de fitxers (en el directori */var/run*) en forma de

fitxer que conté un valor numèric (en text) corresponent al PID del procés. Amb el servei en marxa sempre es pot observar el PID del servidor amb:

```
[root@portatil ~]# ps ax | grep named
 3612 ?        Ss     0:00 /usr/sbin/named

[root@portatil ~]# service named  status
named (pid 3612) s'està executant...

[root@portatil ~]# ll /var/run/named.pid
-rw-r--r-- 1 root root 5 29 jun 14:17 /var/run/named.pid
[root@portatil ~]# cat /var/run/named.pid
3612
```

Un cop iniciat el servei es crea un fitxer de bloqueig o *lock* amb el nom del servei per evitar iniciar-ne una altra instància. Els fitxers de *lock* usualment es troben a */var/lock* i són un simple fitxer de text buit on la seva pròpia existència ja marca que el servei està en marxa. En parar el servei el fitxer s'elimina. Podem observar això fent:

```
[root@portatil ~]# cat /var/lock/subsys/named
[root@portatil ~]# ll /var/lock/subsys/named
-rw-r--r-- 1 root root 0  1 jun 18:26 /var/lock/subsys/named
```

Tots els serveis del sistema normalment es monitoren anotant en fitxers de text un registre de totes les accions que realitzen, són els fitxers coneguts com a fitxers de *log*. Tant es pot utilitzar un fitxer genèric pel sistema com un fitxer independent per a un servei determinat. El servidor DHCP utilitza el fitxer de monitoratge estàndard */var/log/messages*. En aquest fitxer s'enregistra cada cop que el servei s'engega i s'atura.

```
# exemple de missatges en engegar i aturar el servei
[root@portatil ~]# cat /var/log/messages | grep named
Oct  3 17:40:18 localhost named[5532]: starting BIND 9.5.1-P2 -u named
Oct  3 17:40:18 localhost named[5532]: found 2 CPUs, using 2 worker threads
Oct  3 17:40:18 localhost named[5532]: using up to 4096 sockets
Oct  3 17:40:18 localhost named[5532]: loading configuration from ,/etc/named.
    conf'
Oct  3 17:40:18 localhost named[5532]: max open files (1024) is smaller than
    max sockets
(4096)
Oct  3 17:40:18 localhost named[5532]: using default UDP/IPv4 port range:
    [1024, 65535]
Oct  3 17:40:18 localhost named[5532]: using default UDP/IPv6 port range:
    [1024, 65535]
Oct  3 17:40:18 localhost named[5532]: listening on IPv6 interface lo, ::1#53
Oct  3 17:40:18 localhost named[5532]: listening on IPv4 interface lo,
    127.0.0.1#53
Oct  3 17:40:18 localhost named[5532]: automatic empty zone: 127.IN-ADDR.ARPA
... output suprimit ...
Oct  3 17:40:18 localhost named[5532]: running
Oct  3 17:40:34 localhost named[5532]: received control channel command ,stop'
Oct  3 17:40:34 localhost named[5532]: shutting down: flushing changes
Oct  3 17:40:34 localhost named[5532]: stopping command channel on 127.0.0.1#
    953
Oct  3 17:40:34 localhost named[5532]: stopping command channel on ::1#953
Oct  3 17:40:34 localhost named[5532]: no longer listening on ::1#53
Oct  3 17:40:34 localhost named[5532]: no longer listening on 127.0.0.1#53
Oct  3 17:40:34 localhost named[5532]: exiting
```

Sovint l'administrador vol comprovar que els ports que utilitza el protocol DNS estan oberts. En GNU/Linux fàcilment es pot fer una llista dels serveis associats a

cada port mitjançant el fitxer */etc/services*. Algunes utilitats com nmap permeten detectar els ports oberts. Per exemple, podem fer:

```
#Llista dels ports que inclouen alguna referència DNS:
[root@portatil ~]# cat /etc/services  | grep DNS
[root@portatil ~]# cat /etc/services | grep dns
dnsix           90/tcp                          # DNSIX Securit Attribute Token
     Map
dnsix           90/udp                          # DNSIX Securit Attribute Token
     Map
sdnskmp         558/tcp                         # SDNSKMP
sdnskmp         558/udp                         # SDNSKMP
dns2go          1227/tcp                        # DNS2Go
dns2go          1227/udp                        # DNS2Go
menandmice-dns  1337/tcp                        # menandmice DNS
menandmice-dns  1337/udp                        # menandmice DNS
sunscalar-dns   1870/tcp                        # SunSCALAR DNS Service
sunscalar-dns   1870/udp                        # SunSCALAR DNS Service
ddns-v3         2164/tcp                        # Dynamic DNS Version 3
ddns-v3         2164/udp                        # Dynamic DNS Version 3
spw-dnspreload  3849/tcp                        # SPACEWAY DNS Preload
spw-dnspreload  3849/udp                        # SPACEWAY DNS Prelaod
dns-llq         5352/tcp                        # DNS Long-Lived Queries
dns-llq         5352/udp                        # DNS Long-Lived Queries
mdns            5353/tcp                        # Multicast DNS
mdns            5353/udp                        # Multicast DNS
```

De fet, però, el servei DNS sabem que utilitza el port 53 i no apareix en la llista. Si observem el fitxer */etc/services* veurem que s'indica amb el nom *domain*.

```
[root@portatil ~]# cat /etc/services | grep  domain
domain          53/tcp          # name-domain server
domain          53/udp
```

2.8.2 Eines de comprovació d'un servidor DNS

Hi ha diverses eines per comprovar el funcionament d'un servidor DNS, i les més usuals són **nslookup**, **dig** i **host**. El procés de revisió el podríem basar en els apartats següents:

1. Comprovar que el servei està en funcionament.
2. Comprovar un nom local del domini. Provar-lo sense qualificar (només el nom.tipus *pc01*), qualificat amb punt al final (tipus *pc01.escola.org.*) i sense el punt al final (tipus *pc01.escola.org*). Exemples:
   - nslookup server ns1.inf.ioc.cat
   - nslookup server.inf.ioc.cat ns1.inf.ioc.cat
   - nslookup server. ns1.inf.ioc.cat
   - nslookup server.ins.ioc.cat. ns1.inf.ioc.cat
   - nslookup server.ins.ioc.cat ns1.inf.ioc.cat
3. Comprovar un nom de domini remot, fora del domini que s'està provant. Per exemple:

```
nslookup www.uoc.net ns1.inf.ioc.cat
```

4. Comprovar des d'un servidor remot un nom de domini local a la nostra zona. Cal que la zona estigui integrada en l'altre espai de noms. És a dir, cal una zona pare que hagi delegat en el domini que s'està revisant. Per exemple:

nslookup mahatma <nom_ns_de un proveidor ISP>. Si es vol, es pot crear un subdomini a la zona i delegar, i el domini pare ha de permetre resoldre noms del domini fill.

Tot seguit es presenten diverses eines de consulta DNS molt populars. Les utilitats nslookup, dig i host permeten obtenir informació dels recursos d'un servidor DNS, resoldre consultes i saber quins són els seus registres de recurs.

Altres eines que cal saber utilitzar són la utilitat nmap, que permet analitzar els ports d'un *host*; i sniffers de xarxa tipus iptraf i wireshark, que permeten monitorar el trànsit TCP i UDP.

Utilitat dig

Exemple de consulta dels registres de recurs del servidor *ioc.cat*:

```
[root@portatil ~]# dig ioc.cat ANY

; <<>> DiG 9.9.5-9-Debian <<>> ioc.cat ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34042
;; flags: qr rd ra; QUERY: 1, ANSWER: 10, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ioc.cat.     IN  ANY

;; ANSWER SECTION:
ioc.cat.    858 IN  SOA dns1.nominalia.com. root.dns1.nominalia.com. 1431937190
     86400 7200 2592000 300
ioc.cat.    858 IN  MX  10 aspmx.l.google.com.
ioc.cat.    858 IN  MX  30 aspmx2.googlemail.com.
ioc.cat.    858 IN  MX  30 aspmx5.googlemail.com.
ioc.cat.    858 IN  MX  30 aspmx3.googlemail.com.
ioc.cat.    858 IN  MX  20 alt1.aspmx.l.google.com.
ioc.cat.    858 IN  MX  20 alt2.aspmx.l.google.com.
ioc.cat.    858 IN  MX  30 aspmx4.googlemail.com.
ioc.cat.    858 IN  NS  dns2.nominalia.com.
ioc.cat.    858 IN  NS  dns1.nominalia.com.

;; AUTHORITY SECTION:
ioc.cat.    858 IN  NS  dns2.nominalia.com.
ioc.cat.    858 IN  NS  dns1.nominalia.com.

;; ADDITIONAL SECTION:
aspmx.l.google.com. 14  IN  A 173.194.67.26
aspmx.l.google.com. 14  IN  AAAA  2a00:1450:400c:c00::1a
dns1.nominalia.com. 264 IN  A 81.88.57.102
dns2.nominalia.com. 349 IN  A 62.193.205.63

;; Query time: 3 msec
;; SERVER: 147.83.2.3#53(147.83.2.3)
```

```
;; WHEN: Thu Jun 04 16:24:11 CEST 2015
;; MSG SIZE  rcvd: 408
```

En el següent exemple es consulten quins són els servidors de noms arrel:

```
[root@portatil ~]# dig . ANY
; <<>> DiG 9.4.2 <<>> . ANY
;; global options:  printcmd
;; Got answer:
;; —>>HEADER<<— opcode: QUERY, status: NOERROR, id: 47921
;; flags: qr rd ra; QUERY: 1, ANSWER: 14, AUTHORITY: 13, ADDITIONAL: 3

;; QUESTION SECTION:
;.                              IN      ANY

;; ANSWER SECTION:
.                       240358  IN      NS      H.ROOT—SERVERS.NET.
.                       240358  IN      NS      B.ROOT—SERVERS.NET.
.                       240358  IN      NS      I.ROOT—SERVERS.NET.
.                       240358  IN      NS      K.ROOT—SERVERS.NET.
.                       240358  IN      NS      J.ROOT—SERVERS.NET.
.                       240358  IN      NS      C.ROOT—SERVERS.NET.
.                       240358  IN      NS      G.ROOT—SERVERS.NET.
.                       240358  IN      NS      L.ROOT—SERVERS.NET.
.                       240358  IN      NS      E.ROOT—SERVERS.NET.
.                       240358  IN      NS      M.ROOT—SERVERS.NET.
.                       240358  IN      NS      F.ROOT—SERVERS.NET.
.                       240358  IN      NS      D.ROOT—SERVERS.NET.
.                       240358  IN      NS      A.ROOT—SERVERS.NET.
.                       62937   IN      SOA     A.ROOT—SERVERS.NET. NSTLD.
    VERISIGN—GRS.COM.
2007121601 1800 900 604800 86400

;; AUTHORITY SECTION:
.                       240358  IN      NS      L.ROOT—SERVERS.NET.
.                       240358  IN      NS      A.ROOT—SERVERS.NET.
.                       240358  IN      NS      D.ROOT—SERVERS.NET.
.                       240358  IN      NS      J.ROOT—SERVERS.NET.
.                       240358  IN      NS      H.ROOT—SERVERS.NET.
.                       240358  IN      NS      C.ROOT—SERVERS.NET.
.                       240358  IN      NS      I.ROOT—SERVERS.NET.
.                       240358  IN      NS      M.ROOT—SERVERS.NET.
.                       240358  IN      NS      B.ROOT—SERVERS.NET.
.                       240358  IN      NS      G.ROOT—SERVERS.NET.
.                       240358  IN      NS      F.ROOT—SERVERS.NET.
.                       240358  IN      NS      E.ROOT—SERVERS.NET.
.                       240358  IN      NS      K.ROOT—SERVERS.NET.

;; ADDITIONAL SECTION:
A.ROOT—SERVERS.NET.     601520  IN      A       198.41.0.4
B.ROOT—SERVERS.NET.     581337  IN      A       192.228.79.201
C.ROOT—SERVERS.NET.     581337  IN      A       192.33.4.12

;; Query time: 0 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Mon Dec 17 19:02:12 2007
;; MSG SIZE  rcvd: 502
```

El següent és un exemple de consulta dels servidors autoritaris que existeixen per al domini *.com*:

```
[root@portatil ~]#  dig com. ANY
; <<>> DiG 9.4.2 <<>> com. ANY
;; global options:  printcmd
;; Got answer:
;; —>>HEADER<<— opcode: QUERY, status: NOERROR, id: 34936
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 13, ADDITIONAL: 3
```

```
;; QUESTION SECTION:
;com.                           IN      ANY

;; ANSWER SECTION:
com.                    80008   IN      NS      h.gtld-servers.net.
com.                    80008   IN      NS      k.gtld-servers.net.
com.                    80008   IN      NS      b.gtld-servers.net.
com.                    80008   IN      NS      i.gtld-servers.net.
com.                    80008   IN      NS      m.gtld-servers.net.
com.                    80008   IN      NS      l.gtld-servers.net.
com.                    80008   IN      NS      e.gtld-servers.net.
com.                    80008   IN      NS      j.gtld-servers.net.
com.                    80008   IN      NS      f.gtld-servers.net.
com.                    80008   IN      NS      a.gtld-servers.net.
com.                    80008   IN      NS      g.gtld-servers.net.
com.                    80008   IN      NS      c.gtld-servers.net.
com.                    80008   IN      NS      d.gtld-servers.net.

;; AUTHORITY SECTION:
com.                    80008   IN      NS      c.gtld-servers.net.
com.                    80008   IN      NS      j.gtld-servers.net.
com.                    80008   IN      NS      g.gtld-servers.net.
com.                    80008   IN      NS      h.gtld-servers.net.
com.                    80008   IN      NS      k.gtld-servers.net.
com.                    80008   IN      NS      i.gtld-servers.net.
com.                    80008   IN      NS      e.gtld-servers.net.
com.                    80008   IN      NS      l.gtld-servers.net.
com.                    80008   IN      NS      f.gtld-servers.net.
com.                    80008   IN      NS      a.gtld-servers.net.
com.                    80008   IN      NS      b.gtld-servers.net.
com.                    80008   IN      NS      m.gtld-servers.net.
com.                    80008   IN      NS      d.gtld-servers.net.

;; ADDITIONAL SECTION:
a.gtld-servers.net.     72862   IN      A       192.5.6.30
a.gtld-servers.net.     79678   IN      AAAA    2001:503:a83e::2:30
b.gtld-servers.net.     72862   IN      A       192.33.14.30

;; Query time: 0 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Mon Dec 17 19:02:50 2007
;; MSG SIZE  rcvd: 487
```

Utilitat nslookup

Podem utilitzar la utilitat nslookup per fer consultes DNS de tot tipus. Es pot fins i tot seleccionar quin és el servidor al qual es dirigeixen les consultes.

Observeu quins són els nodes arrel:

```
nslookup:
> .
Server:         192.168.0.10
Address:        192.168.0.10#53

Non-authoritative answer:
.       nameserver = G.ROOT-SERVERS.NET.
.       nameserver = F.ROOT-SERVERS.NET.
.       nameserver = L.ROOT-SERVERS.NET.
.       nameserver = I.ROOT-SERVERS.NET.
.       nameserver = E.ROOT-SERVERS.NET.
.       nameserver = K.ROOT-SERVERS.NET.
.       nameserver = A.ROOT-SERVERS.NET.
.       nameserver = C.ROOT-SERVERS.NET.
.       nameserver = M.ROOT-SERVERS.NET.
```

```
.       nameserver = H.ROOT-SERVERS.NET.
.       nameserver = D.ROOT-SERVERS.NET.
.       nameserver = J.ROOT-SERVERS.NET.
.       nameserver = B.ROOT-SERVERS.NET.
.
        origin = A.ROOT-SERVERS.NET
        mail addr = NSTLD.VERISIGN-GRS.COM
        serial = 2007121601
        refresh = 1800
        retry = 900
        expire = 604800
        minimum = 86400

Authoritative answers can be found from:
.       nameserver = J.ROOT-SERVERS.NET.
.       nameserver = F.ROOT-SERVERS.NET.
.       nameserver = K.ROOT-SERVERS.NET.
.       nameserver = I.ROOT-SERVERS.NET.
.       nameserver = C.ROOT-SERVERS.NET.
.       nameserver = D.ROOT-SERVERS.NET.
.       nameserver = M.ROOT-SERVERS.NET.
.       nameserver = G.ROOT-SERVERS.NET.
.       nameserver = E.ROOT-SERVERS.NET.
.       nameserver = A.ROOT-SERVERS.NET.
.       nameserver = B.ROOT-SERVERS.NET.
.       nameserver = L.ROOT-SERVERS.NET.
.       nameserver = H.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET      internet address = 198.41.0.4
B.ROOT-SERVERS.NET      internet address = 192.228.79.201
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
```

Observeu els noms dels servidors autoritaris per al domini *.cat*:

```
nslookup:
> cat
Server:         192.168.0.10
Address:        192.168.0.10#53

Non-authoritative answer:
cat
        origin = ns.nic.cat
        mail addr = dnsmaster.knipp.de
        serial = 2007121773
        refresh = 10800
        retry = 10800
        expire = 604800
        minimum = 86400
cat     nameserver = ns1.nic.es.
cat     nameserver = merapi.switch.ch.
cat     nameserver = ns-ext.nrt1.isc.org.
cat     nameserver = ns.nic.cat.
cat     nameserver = ns-ext.isc.org.
cat     nameserver = ns5.knipp.de.
cat     nameserver = dns4.ad.
cat     nameserver = cat-dns.denic.de.
cat     nameserver = dns-cat.pch.net.

Authoritative answers can be found from:
cat     nameserver = dns4.ad.
cat     nameserver = ns1.nic.es.
cat     nameserver = ns-ext.nrt1.isc.org.
cat     nameserver = ns-ext.isc.org.
cat     nameserver = merapi.switch.ch.
cat     nameserver = cat-dns.denic.de.
cat     nameserver = dns-cat.pch.net.
cat     nameserver = ns.nic.cat.
cat     nameserver = ns5.knipp.de.
ns1.nic.es      internet address = 194.69.254.1
ns5.knipp.de    internet address = 195.253.6.62
```

```
dns4.ad internet address = 194.158.64.10
ns-ext.isc.org  internet address = 204.152.184.64
```

Utilitzant el servidor de noms d'un altre domini, per exemple *dns.gencat.net*, també podem fer-li consultes:

```
nslookup
> server 83.247.128.1
Default server: 83.247.128.1
Address: 83.247.128.1#53
> set type=any
> mail.yahoo.com
Server:         83.247.128.1
Address:        83.247.128.1#53

Non-authoritative answer:
mail.yahoo.com  canonical name = login.yahoo.com.

Authoritative answers can be found from:
yahoo.com       nameserver = ns2.yahoo.com.
yahoo.com       nameserver = ns1.yahoo.com.
yahoo.com       nameserver = ns3.yahoo.com.
yahoo.com       nameserver = ns4.yahoo.com.
yahoo.com       nameserver = ns5.yahoo.com.
yahoo.com       nameserver = ns6.yahoo.com.
yahoo.com       nameserver = ns8.yahoo.com.
ns2.yahoo.com   internet address = 68.142.255.16
ns1.yahoo.com   internet address = 66.218.71.63
ns3.yahoo.com   internet address = 217.12.4.104
ns4.yahoo.com   internet address = 68.142.196.63
ns5.yahoo.com   internet address = 216.109.116.17
ns6.yahoo.com   internet address = 202.43.223.170
ns8.yahoo.com   internet address = 202.165.104.22
```

Utilitat host

Per saber a qui correspon l'adreça IP utilitzada en l'últim exemple anterior:

```
[root@portatil ~]# host 83.247.128.1
1.128.247.83.in-addr.arpa domain name pointer dns.gencat.net.

Per veure la informació del domini de la uoc:
[root@portatil ~]# host -a uoc.es
Trying "uoc.es"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51683
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;uoc.es.                                IN      ANY

;; ANSWER SECTION:
uoc.es.                 86400   IN      NS      nepal.uoc.es.
uoc.es.                 86400   IN      NS      tibet.uoc.es.
uoc.es.                 86400   IN      SOA     tibet.uoc.es. root.tibet.uoc.es
    . 2008062000
28800 7200 604800 86400

;; ADDITIONAL SECTION:
nepal.uoc.es.           86077   IN      A       213.73.40.47
tibet.uoc.es.           86077   IN      A       213.73.40.45

Received 137 bytes from 80.58.61.250#53 in 97 ms
```

Per veure els servidors de correu de *Google*:

```
[root@portatil ~]# host -t MX google.com
google.com mail is handled by 10 smtp1.google.com.
google.com mail is handled by 10 smtp2.google.com.
google.com mail is handled by 10 smtp3.google.com.
google.com mail is handled by 10 smtp4.google.com.
```

Utilitat nmap

La utilitat nmap fa un escaneig dels ports de l'equip indicat. Molt sovint s'ha considerat una eina atacant, però de fet és una excel·lent eina per a un administrador per observar l'estat dels ports del servidor. Podem generar la llista de l'estat del port on escolta el servidor DNS:

```
[root@dnsServer ~]# nmap localhost
Starting Nmap 4.20 ( http://insecure.org ) at 2008-06-07 14:50 CEST
Interesting ports on localhost (127.0.0.1):
Not shown: 1692 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
53/udp   open  dns
80/tcp   open  http
111/tcp  open  rpcbind
443/tcp  open  https
8000/tcp open  http-alt
Nmap finished: 1 IP address (1 host up) scanned in 0.116 seconds
```

Utilitat ping

Podem observar el trànsit UDP d'una consulta DNS amb la utilitat iptraf on es pot veure la connexió del client de sortida usant el port dinàmic 32771 al port 53 del servidor 80.58.61.250. Per una banda activar iptraf i per l'altra fer una consulta que requereixi resolució DNS.

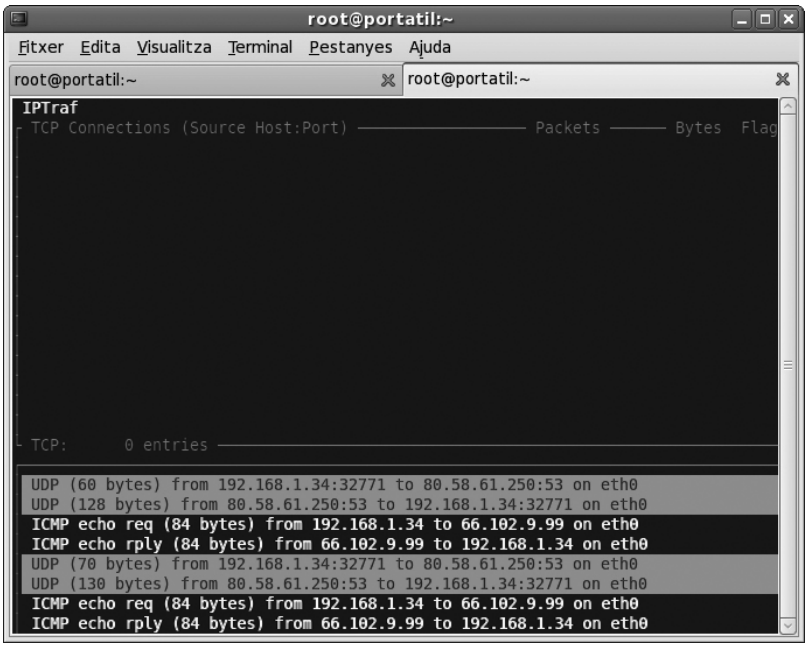
```
[root@portatil ~]# ping www.google.com
PING www.l.google.com (66.102.9.99) 56(84) bytes of data.
64 bytes from 66.102.9.99: icmp_seq=1 ttl=244 time=114 ms
64 bytes from 66.102.9.99: icmp_seq=2 ttl=244 time=74.7 ms

--- www.l.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 74.735/94.539/114.343/19.804 ms
```

En una altra sessió hem fet prèviament:

```
[root@portatil ~]# iptraf
```

La figura 2.6 permet observar el trànsit UDP corresponent a un diàleg DNS.

FIGURA 2.6. Imatge d'una captura amb iptraf



2.8.3 Consultar la configuració del client: el resolver

El client d'una consulta DNS utilitza el *resolver*. Són biblioteques amb les quals està enllaçada l'aplicació client que correspongui. És a dir, no hi ha un programa *resolver* client, sinó que cada aplicació client o el dimoni de xarxa n'implementa un.

El fitxer que conté la configuració del *resolver* en sistemes GNU/Linux és el */etc/resolv.conf*:

```
[root@portatil ~]# cat /etc/resolv.conf
; generated by /sbin/dhclient-script
search local.lan
nameserver 80.58.61.250
nameserver 80.58.61.254
```

L'ordre en què s'han de resoldre les consultes client pot ser primer mirant els fitxers de configuració local (usualment el fitxer */etc/hosts*) i després consultant un servidor DNS o a l'inrevés. Aquesta configuració es defineix en el fitxer */etc/nsswitch*.

```
[root@portatil ~]# cat /etc/nsswitch.conf
#
# /etc/nsswitch.conf
... output suprimit ...
# Example:
#passwd:    db files nisplus nis
#shadow:    db files nisplus nis
#group:     db files nisplus nis

passwd:     files
shadow:     files
group:      files

```

```
#hosts:     db files nisplus nis dns
hosts:      files dns
```

2.8.4 Monitorar el trànsit amb Wireshark

Activar un *snifer* de xarxa com per exemple *Wireshark* per monitorar el trànsit DNS.

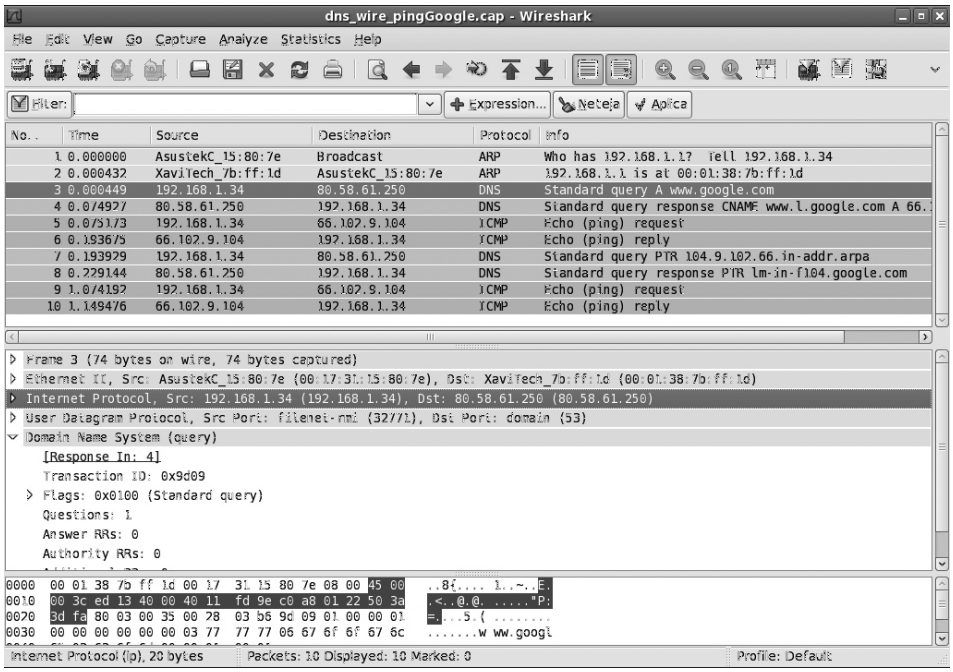
Podem fer que el client sol·liciti una consulta DNS simplement fent un ping a una nova adreça IP que no estigui al *cache* local i que calgui resoldre via DNS.

```
[root@portatil ~]# ping  www.google.com
PING www.l.google.com (66.102.9.104) 56(84) bytes of data.
64 bytes from lm-in-f104.google.com (66.102.9.104): icmp_seq=1 ttl=244 time=118
     ms
64 bytes from lm-in-f104.google.com (66.102.9.104): icmp_seq=2 ttl=244 time
    =75.3 ms
--- www.l.google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 75.313/96.914/118.516/21.603 ms
```

En la captura següent (figura 2.7) podeu observar que s'ha fet el *ping*, el client ha resolt per ARP l'adreça del router local i llavors ha preguntat al servidor DNS (80.58.61.250) per l'adreça de www.google.com. El servidor DNS ha contestat indicant que l'adreça IP buscada és *66.102.9.104*. Fixeu-vos que és una consulta DNS que busca un registre de *tipus A* (*host*).

Podeu observar també que s'ha fet una consulta de resolució inversa en la qual s'ha comprovat que l'adreça IP *66.102.9.104* correspon realment al domini www.google.com.

FIGURA 2.7. Imatge d'una captura de trànsit DNS utilitzant l'aplicació wireshark



A continuació, podeu observar la llista de text de les quatre trames que contenen informació DNS capturades amb el *wireshark* (s'han exportat en format text):

Podeu descarregar-vos aquesta captura del diàleg DNS en la secció "Annexos" del web del mòdul.

```
No. Time        Source          Destination     Protocol   Info
  3 0.000449    192.168.1.34    80.58.61.250    DNS        Standard query A www
      .google.com

Frame 3 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: AsustekC_15:80:7e (00:17:31:15:80:7e), Dst: XaviTech_7b:ff:1d
(00:01:38:7b:ff:1d)
Internet Protocol, Src: 192.168.1.34 (192.168.1.34), Dst: 80.58.61.250
    (80.58.61.250)
User Datagram Protocol, Src Port: filenet-rmi (32771), Dst Port: domain (53)
Domain Name System (query)
    [Response In: 4]
    Transaction ID: 0x9d09
    Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.google.com: type A, class IN

No. Time        Source          Destination     Protocol   Info
  4 0.074927    80.58.61.250    192.168.1.34    DNS        Standard query
      response CNAME
www.l.google.com A 66.102.9.104 A 66.102.9.147 A 66.102.9.99

Frame 4 (142 bytes on wire, 142 bytes captured)
Ethernet II, Src: XaviTech_7b:ff:1d (00:01:38:7b:ff:1d), Dst: AsustekC_15:80:7e
(00:17:31:15:80:7e)
Internet Protocol, Src: 80.58.61.250 (80.58.61.250), Dst: 192.168.1.34
    (192.168.1.34)
User Datagram Protocol, Src Port: domain (53), Dst Port: filenet-rmi (32771)
Domain Name System (response)
    [Request In: 3]
    [Time: 0.074478000 seconds]
    Transaction ID: 0x9d09
    Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 4
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.google.com: type A, class IN
    Answers
        www.google.com: type CNAME, class IN, cname www.l.google.com
        www.l.google.com: type A, class IN, addr 66.102.9.104
        www.l.google.com: type A, class IN, addr 66.102.9.147
        www.l.google.com: type A, class IN, addr 66.102.9.99

No. Time        Source          Destination     Protocol   Info
  7 0.193929    192.168.1.34    80.58.61.250    DNS        Standard query PTR
      104.9.102.66.in-
addr.arpa

Frame 7 (85 bytes on wire, 85 bytes captured)
Ethernet II, Src: AsustekC_15:80:7e (00:17:31:15:80:7e), Dst: XaviTech_7b:ff:1d
(00:01:38:7b:ff:1d)
Internet Protocol, Src: 192.168.1.34 (192.168.1.34), Dst: 80.58.61.250
    (80.58.61.250)
User Datagram Protocol, Src Port: filenet-rmi (32771), Dst Port: domain (53)
Domain Name System (query)
    [Response In: 8]
    Transaction ID: 0x6539
    Flags: 0x0100 (Standard query)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
```

```
    Additional RRs: 0
    Queries
        104.9.102.66.in-addr.arpa: type PTR, class IN

No. Time        Source          Destination     Protocol   Info
  8 0.229144    80.58.61.250    192.168.1.34    DNS        Standard query
      response PTR lm-
in-f104.google.com

Frame 8 (120 bytes on wire, 120 bytes captured)
Ethernet II, Src: XaviTech_7b:ff:1d (00:01:38:7b:ff:1d), Dst: AsustekC_15:80:7e
(00:17:31:15:80:7e)
Internet Protocol, Src: 80.58.61.250 (80.58.61.250), Dst: 192.168.1.34
    (192.168.1.34)
User Datagram Protocol, Src Port: domain (53), Dst Port: filenet-rmi (32771)
Domain Name System (response)
    [Request In: 7]
    [Time: 0.035215000 seconds]
    Transaction ID: 0x6539
    Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
    Queries
        104.9.102.66.in-addr.arpa: type PTR, class IN
    Answers
        104.9.102.66.in-addr.arpa: type PTR, class IN, lm-in-f104.google.com
```

2.9 Realitzar documentació de suport a l'usuari

Una de les facetes més ignorades en el camp de la informàtica és la confecció de manuals i documentació de suport. Com a clients molt sovint ens queixem que falta informació o que està mal redactada. Com a administradors de xarxa no trobem mai temps per anotar les coses. Mentre les tenim al cap no creiem necessari fer la documentació, quan no ho tenim al cap, ja ens és impossible fer-ho, i sovint, és just quan es faria falta haver-ho fet! Fem un repàs de la informació necessària que cal documentar, tant per a l'usuari com per a l'administrador.

El client ha de saber:

- Com contactar amb el servidor DNS. Quin programari ha d'utilitzar i com l'ha de configurar per fer ús del servei.
- Quina és la informació que s'obtindrà via DNS. Cal saber visualitzar, consultar aquesta informació i saber què significa, per a què serveix.

Així doncs, la documentació de l'usuari descriurà el procés per activar el client dns, amb captures de pantalla que facilitin aquest procés. La part més important és mostrar un exemple de configuració de xarxa on es detalli el significat de cada element i explicar a l'usuari com fer aquesta consulta. Un exemple d'informació a proporcionar podria ser el següent:

```
# L'usuari pot consultar la configuració dns l'ordre:
C:\>ifconfig /all
```

```
Configuración IP de Windows
        Nombre del host . . . . . . . . . . : nombre-29b9943f
        Sufijo DNS principal  . . . . . . . :
        Tipo de nodo. . . . . . . . . . . . : híbrido
        Enrutamiento habilitado. . . . . .: No
        Proxy WINS habilitado. . . . .    : No
        Lista de búsqueda de sufijo DNS:    local.lan

Adaptador Ethernet Conexiones de red inalámbricas          :
        Estado de los medios. . . .: medios desconectados
        Descripción. . . . . . . . . . . . : Intel(R) PRO/Wireless 3945ABG Net
            Connection
        Dirección física. . . . . . . . . : 00-31-02-44-9F-5A

Adaptador Ethernet Conexión de área local          :
        Sufijo de conexión específica DNS : local.lan
        Descripción. . . . . . . . . . . . : Realtek RTL8168/8111 PCI-E Gigabit
            Ether NIC
        Dirección física. . . . . . . . . : 00-21-32-80-23-7D
        DHCP habilitado. . . . . . . . . . : No
        Autoconfiguración habilitada. . . : Sí
        Dirección IP. . . . . . . . . . . . : 192.168.1.33
        Máscara de subred . . . . . . . . . : 255.255.255.0
        Puerta de enlace predeterminada   : 192.168.1.1
        Servidor DHCP . . . . . . . . . . . : 192.168.1.1
        Servidores DNS . . . . . . . . . . .: 80.58.61.250
                                              80.58.61.254
        Concesión obtenida . . . . . . . . : jueves, 24 de septiembre de 2009
            12:09:16
        Concesión expira . . . . . . . . . .: jueves, 24 de septiembre de 2009
            13:09:16

# Podem observar els servidors de noms DNS que s'utilitzen.
```

La documentació de l'usuari pot incloure també referències a la documentació oficial del programari client que s'estigui utilitzant i enllaços web d'aquest programari.