10. SERVERUL DNS

10.1. Sistemul DNS (Domain Name System)

10.1.1. Descriere generala

În majoritatea retelelor zilelor noastre, statiile sunt identificate cu ajutorul acestor valori numerice numite adrese IP. Deși ele reprezintă o soluție potrivita pentru schimbul de date intre statii, pentru utilizatorul uman memorarea și utilizarea lor ridica probleme; dacă ne-am limita la adrese, tot internetul ar fi o mare de adrese IP, făcând mult mai dificilă navigarea (asta în cazul în care ar mai fi cunoscut dezvoltarea pe care a căpătat-o până acum). De aceea, încă de la inceputurile retelelor au fost create modalitati de a denumi statiile - de a le identifica prin intermediul unor etichete text, ușor de reținut și reprodus.

Sistemul de operare Unix - și mai apoi Linux - ofera o soluție de atribuire de nume statiilor prin intermediul fisierului */etc/hosts*. În acest fisier text se găsesc corespondente de forma *adresa nume*, ca mai jos:

```
127.0.0.1     localhost
10.0.0.2      c2.infoacademy.net
```

De aceste corespondente țin cont comenzile ce ruleaza în sistemul de operare. Spre exemplu, ne putem referi la stația locala cu numele *localhost*, ne putem conecta prin SSH la *c2.infoacademy.net* în loc de 10.0.0.2 etc.

Deși încă prezentă în sistemele de operare Linux/Unix ale zilelor noastre, soluția */etc/hosts* are doua dezavantaje majore:

- *nu scaleaza bine*. Dacă într-o rețea de 10 statii dorim ca orice stație să se poată referi la celelalte prin intermediul unor nume, va trebui sa actualizam fișierul hosts de pe toate statiile și sa specificam cele 10 corespondente adresa-nume. Dacă apare o a 11-a stație, va trebui sa adaugam o noua inregistrare în fișierul *hosts* al tuturor statiilor!
- *este o soluție pur locala*. Dacă dorim să-i acordam numele www.infoacademy.net unei statii din rețea care ruleaza un server web, vom putea face asta în reteaua locala (modificand fișierul *hosts* al tuturor statiilor) însă cum am putea convinge statiile din internet să facă același lucru? Un nume astfel atribuit nu are vizibilitate în afara retelei

Pe vremea când internetul era abia în formare, exista un fisier *hosts.txt* administrat central, pe care fiecare administrator de rețea trebuia să și-l copieze prin FTP odată la câteva zile. Odată cu fenomenala dezvoltare a internetului, metoda nu mai este fiabilă, în primul rând datorită dimensiunilor fișierului în cauză, și apoi din cauza flexibilității reduse - practic administratorul unei anumite rețele putea opera foarte greu modificări în configurarea DNS, el trebuind să contacteze pentru fiecare schimbare autoritatea centrală, iar acea autoritate ar fi fost îngropată în cereri de adaugare/modificare de informatie.

Pentru a rezolva în mod flexibil și scalabil problema denumirii statiilor în internet a fost creat sistemul DNS (Domain Name System) care pornește de la doua cerinte fundamentale:

- un nume atribuit unei statii trebuie sa poată fi vizibil global. Altfel spus, sistemul intretine iluzia existenței unei unice baze de date, interogabila de clienți aflati oriunde pe glob
- pe de alta parte, aceasta baza de date este administrata distribuit. Practic baza de date este divizata în milioane de fragmente și distribuita pe tot globul, astfel încât fiecare individ/entitate să-și poată administra corespondentele nume-adresa din reteaua proprie (însă acestea sa poată fi vizibile global, conform primei cerinte!)

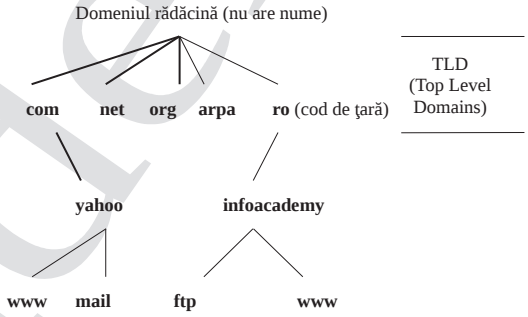
Nota: *deseori, DNS este privit, la modul intuitiv, ca „protocolul care transformă numele în adresă IP”. După cum se va vedea în materialul de fata, sistemul este mai complex de atât, iar baza de date DNS conține și alte tipuri de informație în afara de simplele corespondente nume-adresa.*

Vom imparti prezentarea sistemului DNS în 3 etape:
1. spaţiul de nume DNS şi felul în care este el structurat
2. servere DNS - unde şi cum este păstrată informaţia ce compune marea baza de date DNS
3. clienţi DNS – cine sunt ei și care este modul în care un client DNS poate accesa această informaţie

10.1.2. Spaţiul de nume DNS

10.1.2.1. Structura

În cadrul protocolului DNS nu discutăm despre “adrese” sau “identificatori”, ci despre *nume DNS*. Totalitatea numelor DNS formeaza *spaţiul de nume DNS* (“DNS namespace”). Asemenea adreselor IP, numele DNS formează un spaţiu structurat, şi anume un arbore în care în fiecare nod se află o etichetă, ca în figură, calculatoarele/dispozitivele fiind frunzele arborelui. Fiecare etichetă poate avea până la 64 de caractere; nu sunt permise decât litere, cifre şi semnul “-“ (nu underscore!). Numele unui nod se determină citind etichetele de la el până la rădăcină: www → infoacademy → ro → . Separarea etichetelor în cadrul numelui se face prin punct, aşadar un nume DNS “oficial” este www.infoacademy.ro. Rădăcina (*root*) are ca nume sirul vid.



Observaţie: *putem ușor face analogie intre arborele DNS și sistemul de fisiere, ambele prezentand structuri arborescente. Diferența este ca citirea „căilor” în DNS se face în sens invers (de la frunza către rădăcina).*

O ramură a arborelui care pornește dintr-un anumit nod (adică nodul plus tot ce se află sub el) poarta numele de *domeniu* şi se denumeşte conform nodului în cauzat: de exemplu, domeniul *infoacademy.ro* din figura, care conține nodurile *ftp* și *www*. Un domeniu se poate la rândul său împărţi în *subdomenii*; în afară de rădăcină toate domeniile sunt practic subdomenii.

Pe primul nivel al arborelui se situează domeniile numite *TLD* (Top Level Domains), care pot fi:
- *gTLD* (generic TLD) - cele de uz general, pe care le găsim ca terminaţii la multe dintre site-urile pe care navigăm: **com** (site-uri comerciale), **net** (reţele), **org** (organizaţii), **biz** (bussiness), **info**, **tv**, **arpa** (cu un rol special ce va fi prezentat ulterior)
- *ccTLD* (country code TLD) - etichete de două litere ce reprezintă codul de ţară (ro, tw, uk etc.)

Desi initial lista de TLD-uri era fixa, programul *New gTLD* de la ICANN le permite firmelor sa aplice pentru noi domenii gTLD, imbogatind astfel arborele DNS inca de la primul sau nivel. In plus, a fost introdusa posibilitatea de a folosi in numele DNS caractere din afara setului ASCII, ceea ce deschide calea ccTLD-urilor scrise in limba nativa; spre exemplu, cel pentru Rusia este **.рф**, iar cel pentru Bulgaria **.бг**.

Un nume complet (de la nod/frunză până la rădăcină, și care în consecinta se termina cu punct) este ceea ce se numeşte un **FQDN** (Fully Qualified Domain Name); el este echivalentul caii absolute din sistemul de fisiere. Prin analogie, un nume DNS care nu se termina cu punct este echivalentul unei cai relative. Asa cum o cale relativa primea un prefix - directorul curent - pentru a fi transformata intr-una completa, in acelasi fel, in unele contexte (a se vedea fisierul zona), numele DNS incomplete vor primi un sufix. Aşadar punctul final poate conta, deşi în utilizarea uzuala (ex: adrese scrise în browser) nu ne cramponam de el.

10.1.2.2. Înregistrarea unui domeniu; managementul domeniilor

Înregistrarea unui domeniu este operaţiunea de imbogatire a spatiului de nume DNS prin adaugarea unei noi ramuri, ce are ca efect adugarea de noi inregistrari în baza de date DNS. Pentru realizarea operatiunii este necesar să ştim cine administreaza această bază de date şi cui ne adresăm dacă vrem să cumpărăm un domeniu.

Din acest punct de vedere, autoritatea supremă în internet este ICANN – Internet Corporation for Assigned Names and Numbers; ICANN este o organizaţie non-profit privată care are responsabilitatea managementului global al spaţiului de adrese IP şi al celui de nume DNS, incluzând menţinerea celor 13 servere de root. ICANN a preluat acest job de la IANA – Internet Assigned Numbers Authority, care lucra sub contract cu guvernul USA.

Administrarea este mai departe delegată către autorităţi continentale: RIPE - Réseaux IP Européens, APNIC – Asia/Pacific Information Center, IR pentru America; INTERNIC este autoritatea centrală. Fiecare dintre acestea face delegarea autorităţii pentru subdomenii TLD către *registrars*, adică acele entităţi care interacţionează cu clientul efectuând înregistrarea efectivă a domeniului. Pentru TLD-urile public disponibile (.com, .net, .org etc.) există la ora actuală peste 170 de registrari. Inregistrarea ccTLD-urilor se face de către autorităţile naţionale; pentru informații legate de domeniul *ro* consultati **www.rotld.ro**. În concluzie, pentru a cumpăra/inchiria un domeniu ne adresam unuia dintre registrarii autorizați pentru TLD-ul în cauza.

În funcție de domeniul ales, ne întâlnim cu doua modalitati de a ne fi acordat dreptul de folosinta al domeniului:

- unele domenii se cumpără, posesorul având apoi drept de folosinta pe viața. Este, spre exemplu, cazul domeniilor *.ro* în momentul scrierii acestui material
- alte domenii se închiriază pe perioade intre 1 și 10 ani. Spre exemplu, domeniile de tipul *.com* și *.net*

10.1.3. Servere DNS

10.1.3.1. Zone DNS si rolul serverelor

Serverele DNS indeplinesc urmatoarele roluri in cadrul sistemului:

- stocheaza portiuni ale bazei de date DNS, denumite **zone**. Dupa cum s-a explicat la inceput, baza de date globala DNS (care este de fapt o iluzie intretinuta cu grija) este distribuita pe milioane de servere ce impanzesc globul. Fiecare server gestioneaza una sau mai multe zone DNS
- raspund interogarilor provenite de la clientii DNS. Rostul intregului sistem este ca un client DNS sa poata capata raspunsul dorit, interogand acea baza de date DNS (aparent) centrala; spre exemplu, atunci cand scriem in browser *www.infoacademy.net*, acel nume trebuie sa poata fi transformat in adresa corecta, indiferent unde pe glob ne aflam cand incercam acest lucru si ce server DNS interogam

Zonele reprezinta portiuni ale bazei de date DNS, fiecare corespunzand cate unui subdomeniu din spatiul de nume. Ele contin informatii diverse - atat corespondente nume-IP (ce formeaza serviciul DNS fundamental) cat si multe altele, dupa cum se va vedea in continuarea acestui material. Spre exemplu, zona *yahoo.com* contine inregistrari care stabilesc adresele corespunzatoare numelor *www.yahoo.com*, *mail.yahoo.com*, *messenger.yahoo.com* etc, dar si alte inregistrari care desemneaza serverele e-mail ale domeniului, serverele DNS oficiale etc.

Pe servere, zonele sunt stocate in asa-numitele **fisiere zona**, care sunt fisiere text editabile de catre administrator. Un server care a fost configurat sa stocheze fişierul pentru o anumită zonă se numeşte **autoritativ** pentru zona în cauză (adică este cel responsabil pentru diversele nume DNS din cadrul acelui subdomeniu). Un server autoritativ reprezintă cea mai buna sursa de informație despre acea zona; după cum se va vedea, clientii DNS rareori primesc răspuns direct de la serverele autoritative ale unei zone - în general ei

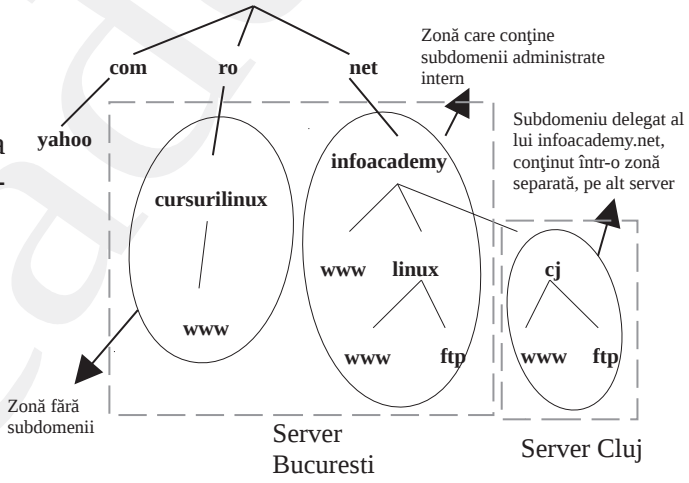
comunica cu servere DNS intermediare. În protocolul DNS exista un flag care ii indica clientului dacă răspunsul primit provine de la un server autoritativ sau de la unul intermediar.

Fiecare domeniu din arborele DNS are un set de servere autoritative, fara de care nu ar putea exista si functiona; sunt incluse in aceasta afirmatie si TLD-urile si domeniul radacina.

10.1.3.2. Conceptul de delegare. Relatia zona-domeniu

Domeniile DNS se cumpara sau se inchiriaza; nu putem cumpara/inchiria un domeniu daca acesta exista deja. Asadar, la cumparare/inchiriere “inventam” un domeniu nou, creand o noua ramura a spatiului de nume DNS.

Sa presupunem ca achizitionam domeniul *infoacademy.net*. La cumparare ni se solicita unul sau mai multe servere DNS; sunt viitoarele servere autoritative pentru zona in cauza. Din momentul in care suntem proprietarii (fie si temporari) ai domeniului si l-am configurat pe serverele autoritative specificate la cumparare, acestea au ca arie de responsabilitate intregul domeniu: avem libertatea de a defini dedesubtul său orice nume de statie (ex: *www.infoacademy.net*, *mail.infoacademy.net* etc), dar si subdomenii noi (ex: *linux.infoacademy.net*, *java.infoacademy.net* etc). In acest moment, pe serverele autoritative ale domeniului este stocata zona *infoacademy.net*, care acopera intregul set de nume al domeniului *infoacademy.net*.



O zona nu corespunde insa intotdeauna unui domeniu integral. Continuand exemplul anterior, sa spunem ca InfoAcademy deschide o filiala in Cluj, care va avea oferte zonale, avand un oarecare grad de independenta fata de filiala centrala, bucuresteana. In consecinta decidem sa-i cream subdomeniul *cj.infoacademy.net*, care va avea propriu-i site (*www.cj.infoacademy.net*), server de mail (*mail.cj.infoacademy.net*), server FTP (*ftp.cj.infoacademy.net*) etc. Nu dorim ca administrarea numelor DNS ale subdomeniului sa se faca la Bucuresti, pe serverul central, ci vrem sa-i acordam filialei posibilitatea de a se autoadministra din punct de vedere DNS. In acest scop vom efectua ceea ce se numeste o **delegare de autoritate**: in loc sa definim subdomeniul *cj.infoacademy.net*, impreuna cu toate numele din cadrul sau, pe serverele centrale (cele responsabile pana acum cu intregul domeniu *infoacademy.net*) vom imputernici serverele DNS ale filialei din Cluj sa se ocupe de administrarea subdomeniului in cauza. Delegarea se realizeaza editand fisierul zona al *infoacademy.net* si specificand noul subdomeniu impreuna cu serverele sale autoritative catre care se face delegarea (vezi 10.2.5.5 pentru modul de implementare a delegarii).

In acest fel am produs un “zone cut” - o linie de separatie intre un domeniu si un subdomeniu al sau. De acum incolo, serverele care administreaza zona *infoacademy.net* nu mai sunt autoritative pentru intregul domeniu *infoacademy.net*, ci doar pentru partea ne-delegata inca.

La modul general, serverele autoritative ale unei zone au ca arie de reponsabilitate toate numele din acea zona, mai putin subdomeniile delegate catre alte servere. Ceea ce am facut noi pentru *cj.infoacademy.net* a facut si *net* pentru noi cand am cumparat *infoacademy.net*: pe serverele responsabile cu domeniul *net* a fost facuta delegarea responsabilitatii pentru subdomeniul *infoacademy.net* catre serverele autoritative ale acestuia din urma. Practic, zona *net* acopera doar o mica parte din domeniul *net*, deoarece majoritatea zdrobitoare a numelor DNS ce au ca terminatie *.net* apartin de subdomenii catre care exista delegare.

Câteva concluzii rapide:

- o zonă DNS este o porţiune a bazei de date DNS care corespunde unei porţiuni de spaţiu de nume ce se intinde de la nodul ce da numele zonei pana la eventualele *zone cuts*
- un domeniu poate avea simultan subdomenii administrate intern (definite în zona de baza ce poarta numele domeniului) şi subdomenii delegate către alte servere
- pe un server pot fi stocate mai multe zone

10.1.3.3. Tipuri de servere DNS asociate unei zone

O zona DNS poate fi gazduita pe unul sau mai multe servere DNS autoritative – unul primar si zero sau mai multe secundare. Serverul primar este cel pe care se creeaza si se poate modifica informatia zonei, pe cand serverele secundare doar isi copiaza zona de pe serverul primar în caz de nevoie.

Nota: *rolurile se pot combina; un soft de server DNS instalat pe o statie poate stoca (gazdui) mai multe zone, jucand rolul de server primar pentru o parte dintre ele si de server secundar pentru celelalte. Nu are sens însă ca un server să fie primar şi secundar pentru aceeaşi zona.*

Iata caracteristici ale celor doua tipuri de servere:

1. **serverul primar** - serverul primar al unei zone este unic si obligatoriu. Este singurul pe care se pot aduce modificari zonei (ex: creare sau modificare de corespondente nume-adresa, adaugare/eliminare roluri de statii etc)

2. **servere secundare** – o zonă poate avea zero sau mai multe servere secundare, şi *este recomandabil* să aibă cel puţin unul. Aceste servere acţionează ca nişte mirror-uri (copii) READ-ONLY ale primarului, sincronizandu-se periodic cu acesta, prin procedeul numit *transfer de zonă*. După cum se va vedea, fiecare zona are o versiune; serverele secundare se conecteaza periodic la serverul primar, verificand dacă versiunea zonei s-a schimbat intre timp, şi nu vor efectua transferul de zona decat daca versiunea de pe serverul primar este superioara. Dacă un secundar nu reuşeşte să-şi actualizeze zona o perioada lunga de timp (configurabila), valabilitatea copiei stocate încetează şi el nu mai este autoritativ pentru zona în cauză

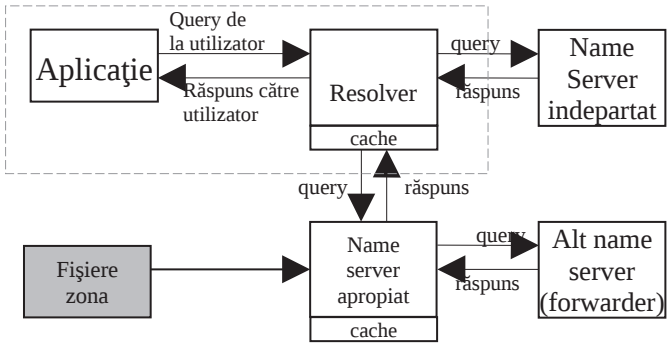
Observatie: *deoarece un server secundar al unei zone are in memoria sa informatia zonei (fie ea si obtinuta prin copiere) el este la rândul sau server autoritativ pentru zona in cauza.*

Prezenta serverelor secundare ofera urmatoarele avantaje:

- **redundanta** – daca serverul primar se afla in imposibilitatea de a raspunde la interogari (defect al statiei, conexiune internet nefunctionala etc), serverele secundare vor raspunde in continuare, asigurand prezenta pe internet a domeniului/domeniilor in cauza
- **load sharing** – serverele secundare, fiind autoritative, raspund şi ele la interogarile legate de zona in cauza, ajutand serverul primar. Sistemul DNS este de asa natura gandit incat sarcina se va distribui in mod egal intre serverele autoritative ale unei zone

10.1.4. Clienti DNS si interogare

10.1.4.1. Cine sunt clientii DNS



Toată această desfăşurare de forţe nu şi-ar avea rostul dacă nu ar exista beneficiarul întregului sistem – clientul DNS. El este acela care vrea să ştie, de exemplu, “cine este www.yahoo.com” - sau mai bine spus care este adresa corespunzatoare acestui nume.

Clientii DNS pot fi:

- *aplicații aflate in sisteme de operare uzuale* (Linux, Windows, Android etc). În astfel de sisteme, foarte multe aplicații au nevoie de servicii DNS și de aceea a fost creat un modul dedicat al sistemului de operare numit *resolver-ul DNS*, de care se folosesc toate aplicatiile. Resolver-ul primeşte din partea aplicaţiei utilizatorului jobul de a „traduce" numele DNS în adrese IP sau invers, contactând în acest scop unul din serverele DNS configurate în sistem. Serverul DNS va trimite fie răspunsul, fie o eroare în cazul în care rezoluţia DNS nu s-a putut efectua.
- *sisteme de operare specializate* - spre exemplu, cele prezente pe echipamente de rețea (routere, switch-uri)
- *alte servere DNS*. După cum se va vedea, serverele se pot folosi unele de altele în procesul de a răspunde la o interogare

Nota: *de la conceptul de resolver vine numele fisierului /etc/resolv.conf din Linux, în care se configureaza serverele DNS de sistem.*

10.1.4.2. Surse de informație ale serverului la primirea unei interogari

Pentru a răspunde la interogari, serverul dispune de mai multe surse de informaţie, a caror ordine de interogare/utilizare poate fi in general stabilita de catre administratorul sistemului de operare:

1. **unul dintre fişierele zonă găzduite** – dacă serverul DNS este autoritativ pentru zona conţinută în interogare, serverul răspunde folosind informaţia locală, fără a avea nevoie de ajutor extern
2. **fişierul *hosts*** – atât în Windows cât şi în Linux/Unix dispunem de acest fişier, în care se pot introduce corespondente statice IP-nume, și care sunt luate în calcul de la bun început de către resolver
3. **memoria cache proprie** – odată ce a obținut răspunsul la o interogare, serverul îl "ţine minte" o perioadă de timp în această memorie, astfel încât poate răspunde rapid dacă i se adreseaza aceeași interogare, minimizând timpii de răspuns şi traficul cu exteriorul
4. **alte servere** – dacă a fost configurat ca atare, un server poate contacta servere DNS ajutatoare pentru a obtine răspunsurile pe care nu le are

Nota: *de remarcat inversiunea temporala intre punctele 3 și 4. Este adevărat ca, dacă un server primește o interogare și are deja raspunsul în cache, îl va livra de acolo fără a mai contacta servere externe; pe de alta parte, un răspuns pătrunde în cache tocmai ca urmare a interogarii unui server extern, atunci când serverul nostru nu cunoaște răspunsul la o interogare adresata de către un client!*

Distingem așadar doua situatii când vine vorba de o interogare adresata de către un client unui server DNS:

- serverul cunoaște răspunsul și i-l livrează clientului. Aici intra cazul în care serverul este autoritativ pentru zona în cauza sau are răspunsul deja disponibil (prezent în cache sau provenit din */etc/hosts*)
- serverul nu cunoaște răspunsul, caz în care are doua posibilități:
  - dacă a fost configurat ca atare, serverul se poate folosi de servere ajutătoare (așa-numitele **forwarders**)
  - dacă nu are forwarders, serverul va urma o procedura ce ii permite aflarea raspunsului pe cont propriu, dar care este mai costisitoare ca timp și resurse consumate (vezi 10.1.4.4). Acesta este și motivul pentru care răspunsul, odată aflat, este memorat în cache pentru refolosire ulterioara

10.1.4.3. Tipuri de interogari

Există două tipuri de interogari DNS:

- interogare **iterativa** – clientul trimite un query către server, solicitandu-i să-i ofere cel mai bun răspuns pe care îl are (fără a mai contacta alte servere). Rezultatul poate fi unul din doua:
  - serverul are deja răspunsul în cache sau într-unul dintre fisierele sale zona

- în cazul în care nu are răspunsul cautat, serverul ii răspunde clientului cu o referinţă (*referral)*, adică indicându-i o listă de servere „cât mai apropiate” de domeniul din întrebare. De exemplu, dacă întrebăm iterativ de www.infoacademy.ro, am putea să primim o listă de servere autoritative pentru „.ro”. De remarcat că în acest caz îi revine clientului sarcina de a continua căutarea răspunsului pe baza referinţelor primite.

- interogare **recursiva** – clientul trimite o singură întrebare, serverul având sarcina de a afla răspunsul chiar dacă asta presupune contactarea altor servere. Altfel spus, clientul „pune la munca” serverul in numele sau. Un server poate fi configurat sa accepte sau nu interogari recursive; spre exemplu, serverele autoritative pentru domenii importante (ex: TLD-urile, domeniul rădăcina etc) nu vor răspunde la astfel de interogari

Rostul unei interogari iterative este de a „deranja” cât mai puțin serverul întrebat, însă de a apropia totuși clientul de răspunsul cautat. După cum se va vedea în continuare, posibilitatea de a comunica cu servere din zona „inalta” a spațiului de nume este vitala pentru rezolutia DNS.

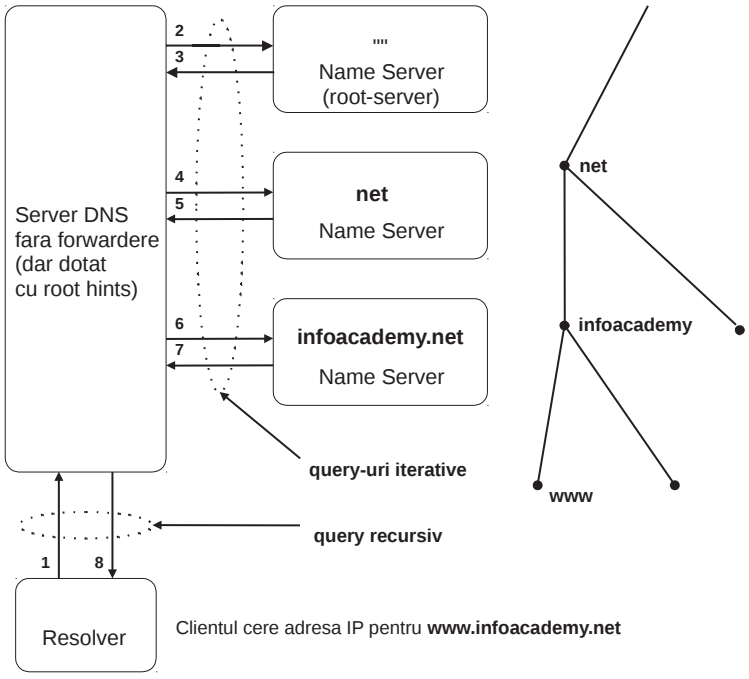
Clientii finali (ex: aplicatiile din sistemele de operare) formuleaza în general interogari recursive; ele sunt adresate unor servere DNS apropiate - cazul tipic fiind cel al serverelor comunicate de providerul de internet. Așadar, în foarte rare cazuri un client va primi răspuns de la un server autoritativ; în general raspunsul fie se va găsi deja în cache-ul serverului interogat, fie acesta din urma va depune efortul aflarii raspunsului în numele clientului.

10.1.4.4. Cazul serverelor DNS fara forwardere

Un server care a primit un query recursiv şi care nici nu este autoritativ pentru zona în cauză, nici nu are răspunsul în cache sau *hosts* și nici nu are forwardere, este nevoit sa descopere răspunsul pe cont propriu. Acest lucru se realizeaza prin parcurgerea de sus în jos a arborelui de nume DNS, mergând pe calea delegarii, pana când este aflat răspunsul corect.

Procedeul se bazează pe faptul ca fiecare nivel cunoaște serverele autoritative ale celor imediat inferioare lui; spre exemplu, serverele autoritative pentru *ro* știu către cine au făcută delegarea în cazul domeniului *infoacademy.ro*. În acest fel, pornind de la rădăcina, putem „alunca” treptat, din nod in nod, urmând delegarile, către serverele autoritative ale domeniului DNS dorit.

Pentru ca procesul sa reușească este necesar sa existe punctul de început: un server lipsit de forwardere are nevoie de o lista de servere de root (cele responsabile cu domeniul rădăcina) - așa-numitele **root hints**. Acestea se pot configura de catre administratorul serverului, iar unele servere vin chiar cu o astfel de lista built-in.



Sa consideram cazul serverului din figura alaturata; pasii de aflare a raspunsului dorit sunt următorii:

1. Clientul solicita adresa IP corespunzatoare numelui www.infoacademy.net
2. Serverul nostru alege unul dintre serverele de root prezente în root hints și ii adreseaza o interogare iterativa, intrebandu-l de *www.infoacademy.net*. Serverul de root nu

cunoaşte răspunsul, dar cunoaşte serverele autoritative pentru domeniul *net*, caci are delegare făcută către ele
3. Serverul de root răspunde cu un referral ce conţine lista de servere autoritative pentru domeniul *net*
4. Serverul nostru alege unul dintre ele şi ii adreseaza aceeaşi interogare, tot iterativa
5. Serverul autoritativ de *net* răspunde cu un referral ce conţine lista de servere autoritative pentru *infoacademy.net*
6. Serverul nostru contacteaza unul dintre serverele autoritative pentru infoacademy.net, adresandu-i aceeaşi întrebare
7. Serverul autoritativ pentru *infoacademy.net* răspunde folosindu-se de informaţia aflata în fişierul său zona
8. Serverul nostru memoreaza răspunsul în cache şi i-l transmite clientului

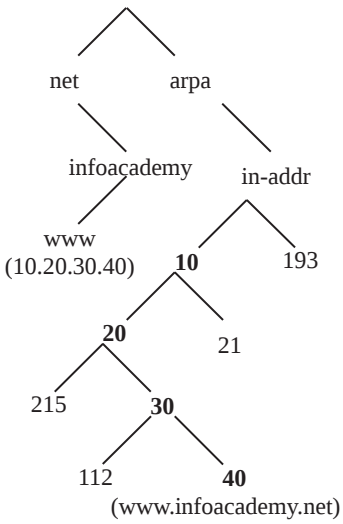
10.1.5. Rezoluţia inversă

Sistemul DNS ofera doua servicii majore:
- *rezolutia DNS directa* - aflarea adresei IP corespunzatoare unui nume DNS
- *rezolutia DNS inversă* - presupune aflarea numelui corespunzător unei adrese IP; aşadar, de data aceasta, informaţia de pornire este adresa

Acestea sunt două operatii total separate: prin DNS nu se realizează *echivalenţe* între adrese IP şi nume, ci doar corespondenţe în câte unul dintre cele doua sensuri (nume→IP sau IP→nume). De aceea în internet nu toate numele care se rezolvă direct beneficiaza şi de rezolutie inversa, ci doar o parte.

Rezoluţia inversă a fost gândită pentru a functiona tot pe structura DNS; acest lucru este facilitat de faptul că adresele IP, la fel ca şi numele DNS, formează o structură ierarhizată, fiind posibilă o punere în corespondenţă a celor doi arbori de identificatori. Există un domeniu DNS alocat special rezoluţiei inverse, şi anume **in-addr.arpa** (*arpa* este TLD-ul, iar *in-addr* vine de la „inverse addressing”). Includerea spaţiului de adrese în cel DNS se bazează pe faptul că prefixul de reţea acţionează ca un fel de domeniu: toate subreţelele sunt incluse în reţeaua mamă, la fel cum subdomeniile sunt incluse în domeniu. „Drumul” către adresa 10.20.30.40 va arăta ca în figura alăturată: dedesubtul domeniului *arpa* se află domeniul 10, care înglobează toate adresele de forma 10.*.*.*. Dedesubt se va afla, printre altele, subdomeniul *20*, corespunzător tuturor adreselor IP de forma 10.20.*.*. Pe nivelul următor se va găsi, printre altele, domeniul *30*, corespunzător adreselor de forma 10.20.30.*, iar printre alte frunze ale acestui domeniu se află şi cea numita 10.0.0.**40**.

Observam aşadar ca nodul ce corespunde adresei 10.20.30.40 se găseşte în zona numita *30.20.10.in-addr.arpa*. Aceasta este o zona în toată regula - este stocata pe un set de servere DNS către care exista delegare din domeniul părinte. Diferenţa fata de o zona obişnuită este ca, pentru nodurile ei, inregistrarile vor fi de tip PTR, nu A (vezi mai jos sectiunea despre tipuri de inregistrari DNS).



În aceste condiţii, pentru rezolutia inversă a adresei 10.20.30.40 utilizam tot o rezolutie directa, însă numele căutat este **40.30.20.10.in-addr.arpa**. Ca regulă generală, pentru rezoluţia inversă a unei adrese x.y.z.t, se cauta inregistrarea de tip PTR corespunzatoare numelui format din adresa pusă invers urmata de sufixul in-addr.arpa: *t.z.y.x.in-addr.arpa*.

Un fapt important (evidentiat şi în figura alaturata) este ca rezoluţia directă şi inversă se realizează pe ramuri diferite ale arborelui, şi deci sunt două operaţii independente, necondiţionându-se în nici un fel una pe cealaltă. De aceea sunt posibile doua situaţii:

- un nume DNS poate avea o adresa corespondenta, însă IP-ul în cauza nu are nume asociat pe ramura de rezolutie inversa
- este posibil ca numele folosit în rezolutia directa sa nu corespunda cu cel din rezolutia inversa! Un scurt exemplu: *www.a.ro* are adresa 185.27.255.3, însă conform rezolutiei inverse numele este *www.romania.org*.

10.2. Serverul DNS BIND (Berkeley Internet Name Domain)

10.2.1. Fisa serverului

Iata pe scurt principalele informatii de pornire, necesare pentru configurarea si administrarea serverului BIND:

- executabil: **named** (“name daemon”)
- fisier de configurare: **named.conf**, aflat de obicei in */etc* sau */etc/bind*
- pornire in foreground, cu logurile afisate pe ecran: **named -g**
- utilitar de control: **rndc** (remote name daemon control). Acesta ofera posibilitatea controlarii serverului de la distanta sau de pe aceeasi masina
- utilitar de validare a fisierului de configurare: **named-checkconf**
- utilitar de validare a fisierelor zona: **named-checkzone**
- utilitare de interogare manuala: **dig, host, nslookup**

10.2.2. Roluri si moduri de configurare ale unui server DNS

Un soft de server DNS se configureaza diferit in functie de scopul sau. Descriem aici urmatoarele roluri ale unui server:

- **server autoritativ** pentru unul sau mai multe domenii. Scopul unui astfel de server este sa asigure prezenta pe internet a domeniilor de care este responsabil. El nu trebuie sa raspunda la interogari recursive, deoarece:
  - nu raspunde decat la interogari legate de domeniile gazduite
  - toate raspunsurile la interogari legate de acele domenii vor proveni din informatia cuprinsa in fisierele sale zona, si deci serverul nu joaca niciodata rolul de intermediar pentru zonele in cauza
- **caching name server** – este un server creat pentru a raspunde la interogarile recursive ale unui set bine delimitat de clienti. Serverul “munceste” in numele acestor clienti, obtinand raspunsuri la interogarile lor si memorandu-le temporar in cache-ul sau, astfel incat o aceeasi interogare repetata isi va primi raspunsul mult mai rapid odata ce acesta a fost retinut in cache. Un caching name server nu va fi autoritativ pentru nicio zona dar va permite interogarile recursive – insa numai de la clientii pe care intentioneaza a-i servi

Rolurile sunt cumulabile: un server DNS poate fi configurat ca server primar pentru cateva zone, server secundar pentru alte cateva, si in plus poate juca rolul de caching name server pentru un set de clienti. Combinarea rolului de server de cache cu cel de server autoritativ se intalneste uneori in firme mici sau in retele “de casa”, unde aceeasi instanta de BIND joaca rolul de caching name server pentru clientii retelei si, in paralel, asigura prezenta pe internet a domeniului/domeniilor dorite. Din motive de comoditate si buget, deseori statia in cauza are si alte roluri (server web, server e-mail, file server etc).

10.2.3. Elemente componente ale configurarii

In functie de rolul jucat de serverul BIND, configurarea sa poate presupune doua aspecte:

1. Editarea fisierului de configurare al serverului, **named.conf**. Acesta contine configurarea daemon-ului *named* (adrese si porturi pe care asculta, restrictii in privinta clientilor care il pot interoga etc). In cazul unui server autoritativ, aici este specificata si lista de zone gazduite pe acest server

2. Editarea fisierelor zona pentru adresarea directa si/sau inversa. Acest pas este necesar numai atunci cand serverul joaca rolul de server primar pentru una sau mai multe zone DNS; in cazul unui caching name server sau al unui server secundar aceasta etapa nu este necesara. Locatia fisierelor zona se stabileste de catre administrator, din *named.conf*.

Asadar, prima operatie (editarea fisierului *named.conf*) este intotdeauna necesara, pe cand cea de-a doua (editare fisiere zona) are loc numai cand BIND joaca rolul de server primar pentru una sau mai multe zone.

10.2.4. Fisierul named.conf

10.2.4.1. Compozitie

10.2.4.1.1. Tipuri de directive

Fisierul *named.conf* contine directive de configurare de doua feluri:

- **directive simple** – sunt directive care seteaza parametri de configurare ai serverului si care in general ocupa o singura linie. Exemplu: activarea recursivitatii, tipul unei zone, calea catre fisierul zona etc.
- **directive bloc** – sunt gandite sa incorporeze una sau mai multe directive simple sau bloc, restrictionand astfel efectul acestora. Spre exemplu, pentru un server autoritativ exista declaratii de zona in cadrul carora se specifica, prin directive simple, setarile ce tin exclusiv de zona in cauza.

Exemplu: o directiva *zone* ce contine doua directive simple, efectul acestora din urma manifestandu-se doar pentru zona specificata:

```
zone “infoacademy.ro”{
      type master;
      file “/var/named/infoacademy.ro”;
};
```

10.2.4.1.2. Sintaxa generala

Fisierul *named.conf* este guvernat de urmatoarele reguli de sintaxa:

- portiunile de linie care incep cu # reprezinta comentarii si sunt ignorate de catre server
- orice directiva – fie ea simpla sau bloc – se incheie cu ; (punct si virgula). ***In cazul directivelor bloc, nu uitati sa puneti simbolul ; dupa acolada inchisa!***
- in cadrul unei directive bloc, directivele simple se pot specifica una sau mai multe pe aceeasi linie. Urmatoarele doua secvente sunt echivalente:

```
allow-query { 10.0.0.5; 192.168.0.5; };
# … echivalent cu a scrie:
allow-query {
      10.0.0.5;
      192.168.0.5;
};
```

10.2.4.1.3. Directive de configurare

Iata cateva dintre cele mai importante directive ce pot fi folosite in fisierul *named.conf*:

- **controls** – defineste canale de control folosite pentru administrarea serverului prin intermediul *rndc*
- **acl** – cu ajutorul acestei directive, atunci cand o aceeasi lista de adrese intervine in mai multe locuri din fisierul de configurare, administratorul ii poate atribui acestei liste un nume cu care o va referi ulterior in cadrul fisierului

- **options** – configureaza parametrii globali sau default ai serverului:
  - acceptarea sau interzicerea globala a interogarilor recursive: **recursion**
  - lista de servere ajutatoare: **forwarders**
  - restrictionari:
    - adrese de la care sunt acceptate query-uri: **allow-query**
    - adrese de la care sunt acceptate query-uri recursive: **allow-recursion**
    - adrese care au dreptul de a primi raspuns din cache: **allow-query-cache**
    - adrese catre care este permis transferul de zona: **allow-transfer**
- **zone** – declara o zona pentru care serverul este autoritativ (primar sau secundar)

Vom detalia in continuare modul de utilizare a acestor directive in functie de scopul serverului.

10.2.4.2. Configurare

10.2.4.2.1. Conectivitatea cu utilitarul de control rndc

Utilitarul de control al serverului, **rndc**, pemite o sumedenie de operatii utile, motiv pentru care dorim sa ne asiguram de corecta lui functionare si interactiune cu serverul. Configurarea sa minimala presupune doua aspecte:

- configurarea serverului astfel incat sa deschida un port pentru conexiunea cu *rndc*. Se realizeaza folosind in *named.conf* directiva bloc **controls** ce specifica adresa si portul pe care asteapta serverul conexiuni de la *rndc*:

```
# in fisierul named.conf:
controls{
      inet 127.0.0.1 port 953 allow { 127.0.0.1; };
}
```

  Dupa cuvantul cheie *inet* poate urma o adresa IP (serverul va asculta numai pe adresa specificata) sau * (serverul va asculta pe toate adresele definite in sistem). Dupa cuvantul cheie *allow* urmeaza o lista de adrese/subnet-uri/ACL-uri (vezi 10.2.4.2.7) care specifica de pe ce masini se permite conectarea lui *rndc* la server.

- securizarea comunicatiei intre server si *rndc* folosind o cheie de criptare simetrica. Aceasta se poate realiza in multiple moduri; prezentam in acest material o modalitate simpla, bazata pe setari implicite - si anume generarea fisierului **/etc/rndc.key** (care este folosit automat de catre server si *rndc*) cu ajutorul utilitarului **rndc-confgen**:

```
rndc-confgen -a
```

Odata ce am efectuat aceste setari, dupa ce am pornit serverul putem folosi utilitarul *rndc* pentru a-l controla (vezi 10.2.6.4).

10.2.4.2.2. Configurarea unui caching name server

Un caching name server nu este autoritativ pentru nicio zona; el doar raspunde interogarilor recursive ale unui set bine delimitat de clienti. El poate fi configurat ca, atunci cand nu are raspunsul la o interogare, sa se foloseasca de alte servere DNS - asa-numitele *forwarder*-e.

Minimul necesar pentru un astfel de server este o directiva bloc **options** ce stabileste restrictiile amintite si configureaza lista de forwardere:

```
# fisierul named.conf
options {
      forwarders { 8.8.8.8; 8.8.4.4; };
      allow-query { 10.0.0.0/24; 192.168.0.0/24; };
      allow-recursion { 10.0.0.0/24; 192.168.0.0/24; };
      # daca am fi pus recursion yes; activam global recursivitatea ceea ce nu era dezirabil
};
```

Nota: *recursivitatea poate fi controlata in doua moduri:*

- *cu directiva* ***recursion*** *(valori posibile yes sau no), care activeaza sau dezactiveaza global recursivitatea*
- *cu directiva* ***allow-recursion****, ca mai sus, specificand setul de clienti pentru care serverul onoreaza interogari recursive. Directiva* allow-recursion *are prioritate fata de* recursion *in cazul in care sunt specificate amandoua.*

10.2.4.2.3. Configurarea unui server primar

Restrictii

Ca si in cazul unui caching name server, administratorul poate stabili restrictii asupra categoriilor de interogari si/sau clienti pentru care raspunde serverul. Deoarece serverul nu raspunde decat la interogari legate de domeniile pentru care este autoritativ, si face acest lucru folosind numai fisierele zona proprii, avem urmatoarele particularitati fata de cazul anterior:

- recursivitatea va fi dezactivata
- nu mai exista forwarder-e
- serverul ii va raspunde oricarui client care il interogheaza

```
options {
      recursion no;              #alternativ puteam scrie allow-recursion { none; };
      allow-query { any; };
};
```

Nota: *cuvintele cheie none si any sunt acl-uri predefinite (vezi 10.2.4.2.7); none are semnificatia "nicio statie" iar any inseamna "orice statie".*

In exemplul de mai sus, recursivitatea este dezactivata, serverul raspunzand tuturor interogarilor primite - indiferent de IP-ul sursa - folosind numai informatia din fisierele zona.

Specificarea zonelor pentru care serverul este autoritativ

Fisierul *named.conf* trebuie sa contina cate o directiva bloc **zone** corespunzatoare fiecarei zone pentru care serverul este autoritativ (primar sau secundar). Iata minimul de informatii care trebuie specificat:

- tipul zonei – mai bine spus, rolul pe care il joaca serverul pentru acea zona. Se stabileste cu directiva **type** dandu-i valoarea *master*
- locatia fisierului zona – este aleasa de catre administrator, cu singura restrictie ca fisierul sa fie accesibil procesului server DNS. Se stabileste cu directiva **file** urmata de calea catre fisierul in cauza

```
# in fisierul named.conf al serverului primar
zone "a.ro"{
      type master;                  # rol de server primar pentru aceasta zona
      file "/var/named/a.ro";       # locatia fisierului zona
};
```

În cazul serverului primar, fișierul zona trebuie creat și populat de către administrator pentru ca serverul sa pornească și sa funcționeze corect.

10.2.4.2.4. Configurarea unui server secundar

In cazul unui server secundar, restrictiile se configureaza in acelasi fel ca pe serverul primar. In schimb in privinta zonei avem urmatoarele diferente:

- zona nu mai este de tip *master*, ci de tip **slave**
- serverul trebuie sa dispuna de o lista de servere de pe care sa-si copieze zona cand este nevoie; ea se stabileste cu directiva bloc **masters**. In lista de servere master se poate afla nu numai serverul primar al zonei, ci si alte servere (vezi 10.2.4.2.5)
- desi se poate specifica locatia fisierului zona, acesta **nu trebuie editat de catre administrator**, deoarece el se populeaza in urma operatiei de transfer de zona

```
# in fisierul named.conf al serverului secundar
zone “a.ro”{
      type slave;                          # rol de server secundar
      file “/var/named/a.ro-sec”;          # aici va fi salvat continutul zonei dupa transfer
      masters { 10.0.0.2; 10.0.0.3; };     # de pe aceste servere este copiata zona
};
```

Atenţie! Nu este suficient sa configuram un server secundar ca slave și el să-și facă corect transferul de zona! Dacă ne limitam la atât, am creat un server „clandestin” - nimeni nu știe de existența lui, și implicit el nu primește interogari și deci nu își realizeaza beneficiile de redundanta și load sharing. Pentru ca serverul sa devină „oficial”, trebuie declarat ca server autoritativ în fișierul zona (vezi 10.2.5).

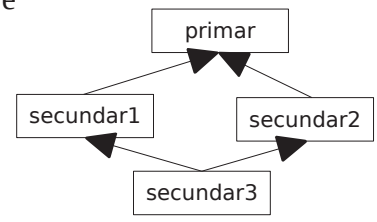
10.2.4.2.5. Clarificari in privinta conceptelor de master si slave, respectiv primar si secundar

Cand spunem ca o zona este “de tip master” sau “de tip slave”, aceasta nu descrie cu adevarat “tipul” zonei ci, de fapt, felul in care actioneaza serverul nostru pentru zona in cauza si implicit modul lui de configurare:

- daca zona este declarata *type master*, administratorul creaza si populeaza manual zona
- daca zona este declarata *type slave*, continutul ei va fi preluat de pe alt server, prin transfer de zona

Asadar, serverul primar al unei zone va avea zona declarata ca *master*, iar cele secundare vor avea zona declarata ca *slave*.

Fiecare server slave trebuie sa aiba unul sau mai multe servere “sursa” de pe care isi copiaza zona in caz de nevoie. Desigur ca, in cazul cel mai simplu si mai firesc, lista de surse ar include un singur server - primarul zonei; insa exista situatii in care un server secundar, din varii motive, nu poate contacta primarul direct, si atunci exista posibilitatea ca el sa-si copieze zona de pe unul dintre celelalte servere secundare. Asadar, un server specificat in declaratia *masters* a unui slave poate sa NU fie serverul primar al zonei.



Sa consideram exemplul din figura alaturata, in care *secundar3* nu are acces direct la serverul primar si isi va copia zona de pe *secundar1* sau *secundar2*. In aceste conditii:

- pe primar zona este configurata ca fiind *type master*
- pe *secundar1* si *secundar2* zona va fi declarata *type slave*, iar in directiva *masters* va aparea doar serverul primar
- pe *secundar3* zona va fi declarata *type slave*, iar in directiva *masters* vor fi listate adresele lui *secundar1* si *secundar2*

10.2.4.2.6. Transferul de zona

Serverele secundare se conecteaza periodic la serverul primar (sau la cel configurat ca sursa a zonei, prin declaratia *masters*) si verifica daca versiunea zonei de acolo este numeric superioara celei de pe serverul initiator; in caz afirmativ se demareaza transferul de zona. Perioada la care se efectueaza verificarea se regleaza prin intermediul inregistrarii SOA din fisierul zona (vezi 10.2.5.2).

Aceasta inseamna ca, daca zona a fost actualizata pe primar, in mod normal secundarele ar trebui sa astepte pana la scurgerea perioadei si abia apoi isi vor da seama de modificare si se vor actualiza. Procesul poate fi urgentat in doua moduri:

- protocolul DNS dispune de un mecanism de notificare (suportat si de BIND), prin care orice server autoritativ care a detectat o modificare a zonei va notifica pe celelalte servere. Serverele care au deja cea mai noua versiune vor ignora notificarea. Notificarea se poate transmite in doua cazuri:
  - ca urmare a unei modificari efectuate de catre administrator pe primar si detectate de catre acesta din urma
  - administratorul poate forta trimiterea notificarilor folosind **rndc notify** pe serverul master (vezi 10.2.6.4)
- administratorul poate forta transferul unei anumite zone:
  - folosind **rndc retransfer** pe serverele slave (vezi 10.2.6.4)
  - folosind **dig** si tipul de RR *AXFR*, ce corespunde transferului de zona (vezi 10.2.6.3)

Transferul de zona poate fi restrictionat pe serverul primar astfel incat sa fie permis numai catre serverele de drept, folosind directiva **allow-transfer**:

```
zona “a.ro”{
      type master;
      file “/var/named/a.ro”;
      allow-transfer { 10.0.0.200; 10.0.0.300; };    # IP-urile secundarelor oficiale
};
```

10.2.4.2.7. ACL-uri

Un ACL reprezinta o modalitate de a-i atribui un nume unei liste de adrese sau subnet-uri, astfel incat sa putem refolosi comod acea lista in mai multe locuri din fisierul de configurare. Un ACL se defineste cu ajutorul directivei **acl** urmata de o insiruire de adrese sau subnet-uri, incadrate de acolade si incheiate fiecare cu punct si virgula:

```
acl “clientii_mei” { 10.0.0.0/24; 192.168.0.0/24; };
options{
      allow-query { “clientii_mei”; };
      allow-recursion { “clientii_mei”; };
};
```

ACL-urile trebuie definite inainte de a fi folosite; din acest motiv, deseori definitiile lor sunt plasate la inceputul fisierului de configurare.

In BIND exista cateva acl-uri predefinite: **none** (nicio statie), **any** (orice statie), **localhost** (orice adresa locala a serverului), **localnets** (toate subnet-urile direct conectate la server).

10.2.5. Fisierele zona

10.2.5.1. Tipuri de inregistrari

Atunci cand BIND joaca rolul de server primar pentru una sau mai multe zone, intervine o etapa suplimentara a configurarii – crearea si popularea fisierelor zona.

Fisierul zona, în calitatea sa de porţiune a bazei de date DNS, reprezintă un ansamblu de înregistrări de diverse tipuri; ele sunt numite **RR – Resource Records**. Iata cele mai importante tipuri de RR:

SOA	Start Of Authority – specifica serverul primar al zonei si reglementeaza relația cu secundarele
NS	Name Server – declară serverele DNS autoritative; folosită şi pentru delegări
A	Address – specifică adresa corespunzatoare unui nume DNS (folosită în rezolutia DNS directa,nume → IP)
AAAA	IPv6 address – specifica adresa IPv6 corespunzatoare unui nume DNS
PTR	Pointer – tip de RR folosit în rezoluţia DNS inversă (IP → nume)
MX	Mail Exchanger – declară un server de mail (SMTP) pentru zona în cauză
CNAME	Canonical Name – defineşte un alias (nume secundar) al unei alte inregistrari

Vom prezenta in continuare sintaxa acestor inregistrari si felul in care trebuie ele imbinate intr-un fisier zona.

10.2.5.2. Formatul unei inregistrari

Formatul general al unei inregistrari este:

```
nume          <TTL>          <CLASA>        TIP_RR         DATE_SPECIFICE
```

Campurile TTL (Time to Live) si CLASA sunt optionale. TTL specifica timpul maxim de validitate al RR-ului in cache-ul clientilor (in secunde), iar CLASS la ora actuala este IN (internet).

Portiunea finala a inregistrarii, DATE_SPECIFICE, ia forma unei succesiuni de campuri a carei lungime si compozitie depinde de tipul inregistrarii. Iata exemple de format al principalelor tipuri de inregistrari:

Format in fisierul zona	Exemplu	Explicatie
`nume A adresa_IP`	`www.x.ro. A 10.0.0.1`	asociaza o adresa IP unui nume
`domeniu NS server_autoritativ`	`x.ro. NS ns1.x.ro.`	specifica un name server autoritativ pentru un domeniu sau face o delegare, in functie de caz
`domeniu MX prioritate nume`	`x.ro. MX 10 m1.x.ro.`	declara un server de mail pentru domeniul specificat
`numar PTR nume`	`12 PTR Comp12.x.ro.`	asociaza un nume unei adrese IP (folosit in rezolutia inversa)
`nume1 CNAME nume`	`ftp.x.ro. CNAME www.x.ro.`	declara nume secundar pentru o statie deja definita in cadrul zonei

Discutam separat inregistrarea de tip SOA deoarece ea este mai complexa. Formatul sau general este:

```
numezona      SOA     serverprimar    email     serial     refresh    retry     expire      minimumTTL
```

Inregistrarea de tip **SOA** are doua roluri:

- specifica serverul primar pentru zona in cauza – singurul pe care se vor putea aduce modificari zonei. El trebuie sa fie unul dintre serverele autoritative prezente sub forma de inregistrari NS
- stabileste parametrii relatiei primar-secundare

Parametrii SOA au urmatoarele semnificatii:

- **serverprimar** este numele DNS al serverului primar al zonei; trebuie sa se regaseasca si in lista de inregistrari NS ale zonei
- **email** - reprezinta adresa administratorului de domeniu, in care primul caracter **.** neprecedat de \ tine loc de @ (deoarece @ este caracter rezervat), si orice alt caracter **.** ce-l precede trebuie reprezentat sub forma **\.**
- **serial** este versiunea zonei; pe baza ei, serverele *slave* decid daca este sau nu necesar transferul de zona
- **refresh** – perioada (exprimata in secunde) la care slave-urile incearca sa-si actualizeze copia zonei
- **retry** – daca transferul de zona esueaza la un moment dat, slave-urile vor re-incerca periodic sa-si sincronizeze copia zonei, repetitia efectuandu-se cu o perioada diferita, data de timpul de *retry*
- **expire** – daca actualizarea zonei nu s-a realizat in acest interval de timp, serverul secundar abandoneaza incercarile si inceteaza a mai fi autoritativ pentru zona in cauza
- **minimum TTL** – specifica TTL-ul default, care se va aplica tuturor inregistrarilor care nu au TTL propriu (specificat in cadrul inregistrarii)

Observam ca numai unul dintre tipurile de inregistrari creeaza corespondente directe nume-adresa; restul specifica roluri ale statiilor ce au nume deja definite.

10.2.5.3. Compozitia de baza a unui fisier zona

O zona are:

- un RR de tip **SOA** *si numai unul*, care este obligatoriu, plasat la inceputul zonei
- cate un RR de tip **NS** pentru fiecare dintre serverele autoritative ale zonei (atat primar cat si secundare)
- eventuale RR-uri de alte tipuri. Spre exemplu, daca numele serverelor autoritative ale zonei fac parte din zona respectiva, zona trebuie sa cuprinda si inregistrari de tip A corespunzatoare lor

Iata un exemplu de fisier zona minimal pentru domeniul *x.ro*:

```
x.ro.                SOA     nsprimar.x.ro.       admin\.dns.x.ro.   (
                                    1             ; serial
                                    3600          ; refresh
                                    60            ; retry
                                    7200          ; expire
                                    60            ; minimum TTL
                               )

                     NS      nsprimar.x.ro.
nsprimar.x.ro.       A       1.2.3.4
```

Nota: *caracterul ; reprezinta inceput de comentariu, iar parantezele rotunde sunt folosite numai atunci cand dorim sa scriem o inregistrare pe mai multe linii, pentru claritate (vezi 10.2.5.4)*

Numele *nsprimar.x.ro.* se repeta de 3 ori in exemplul de mai sus deoarece de fiecare data participa cu o alta semnificatie:

- inregistrarea de tip A creeaza numele *nsprimar* ce are asociata adresa specificata
- inregistrarea NS atribuie statiei *nsprimar.x.ro.* rolul de server autoritativ pentru zona *x.ro*
- inregistrarea SOA precizeaza care dintre serverele autoritative (cele listate sub forma de inregistrare NS) este serverul primar al zonei

Trebuie inteles ca acesta este un fisier zona minimal, ce constituie baza pentru serviciile DNS dorite de fapt de catre posesorul domeniului; de aici incolo se adauga inregistrarile corespunzatoare acestor servicii – iata exemple:

```
; server web
www.x.ro.     A      1.2.3.4
; server FTP, pe aceeasi masina
ftp.x.ro.     CNAME  www.x.ro.
; server email
x.ro.  MX     10     mail.x.ro.
mail.x.ro.    A      5.6.7.8
; server DNS secundar
x.ro.         NS     ns2.x.ro.
ns2.x.ro.     A      9.0.1.2
```

10.2.5.4. Reguli de sintaxa in fisierul zona

Exista cateva reguli de care trebuie tinut cont la populatea fisierului zona:

- orice nume DNS care nu se termina cu **.** este considerat a fi unul relativ si i se adauga ca sufix implicit numele zonei. In consecinta:
  - **nu uitati punctul de final pentru numele complete!**
  - putem folosi nume scurte, bazandu-ne pe faptul ca serverul le va adauga automat sufixul
- caracterul **;** (punct si virgula) reprezinta debutul unui comentariu de o linie
- daca in primul camp al inregistrarii, in locul numelui DNS, se lasa spatiu sau TAB, va fi mostenit automat numele inregistrarii anterioare
- caracterul **@** este substituit automat cu sufixul curent (implicit numele zonei)
- atunci cand dorim sa distribuim o inregistrare pe mai multe linii, deschidem o paranteza rotunda la finalul primei linii si o inchidem la finalul inregistrarii, dupa ultima linie (vezi exemplul de zona din sectiunea anterioara)

Folosindu-ne de aceste reguli putem rescrie fisierul zona de mai sus intr-o forma mult simplificata:

```
@             SOA     nsprimar      admin\.dns   1 3600 60 7200 60
              NS      nsprimar
nsprimar      A       1.2.3.4
```

10.2.5.5. Delegarea unui subdomeniu

Impartirea bazei de date DNS in zone (portiuni stocate si administrate descentralizat) se face pe principiul delegarii de autoritate. Spre exemplu: un server poate stoca numai primul nivel al domeniului *infoacademy.ro*, iar pentru subdomeniul *linux.infoacademy.ro* poate face o delegare – desemneaza alte servere pe care le imputerniceste sa se „ocupe” de acea zona.

In fisierul zona, delegarea ia urmatoarea forma:

```
nume_subdomeniu     NS     nume_nameserver1
nume_subdomeniu     NS     nume_nameserver2
; se vor lista toate serverele autoritative pentru acel domeniu
```

Daca numele serverelor catre care se face delegarea fac parte chiar din subdomeniu, se creeaza un cerc vicios - ca sa se poata ajunge la acele servere este necesara adresa lor, dar care este definita chiar pe serverele in cauza! In astfel de cazuri trebuie adaugate ”glue records” - adica RR-uri de tip A care asociaza adrese celor numelor de servere autoritative aflate in subdomeniu, insa plasate in domeniul parinte (cel care deleaga):

```
nume_nameserver1    A      adresa1
nume_nameserver2    A      adresa2
```

Exemplu: delegarea domeniului *sub.x.ro* catre doua servere din cadrul acestui subdomeniu:

```
x.ro.                SOA    nsprimar.x.ro.       admin\.dns.x.ro. 1 3600 60 7200 60
                     NS     nsprimar.x.ro.
nsprimar.x.ro.       A      1.2.3.4
; delegarea domeniului sub.x.ro
sub.x.ro.            NS     ns1.sub.x.ro.
sub.x.ro.            NS     ns2.sub.x.ro.
; serverele catre care se face delegare au nume din subdomeniu, deci necesita glue records
ns1.sub.x.ro.        A      3.4.5.6
ns2.sub.x.ro.        A      7.8.9.0
; in mod normal locul acestor doua inregistrari era in zona corespunzatoare subdomeniului
```

10.2.5.6. Zonele de rezolutie inversa

O zona de rezolutie inversa se configureaza pana la un punct la fel ca una de rezolutie directa:

- trebuie sa fi fost delegata de serverele domeniului parinte catre serverul pe care se face configurarea ei
- poate fi gazduita pe un server primar si zero sau mai multe secundare
- contine aceleasi tipuri de inregistrari de pornire (SOA, NS)

Diferenta majora este ca inregistrarile suplimentare fata de cele de baza vor fi de tip PTR, deoarece ele realizeaza corespondenta inversa, IP → nume.

Iata un exemplu de configurare de zona de rezolutie inversa pentru adresele IP din reteaua 10.0.0.*:

```
# fisierul named.conf
zone “0.0.10.in-addr.arpa”{
      type master;
      file “/var/named/10.0.0.rev”;
}
```

```
; fisierul zona 10.0.0.rev
0.0.10.in-addr-arpa.       SOA    nsprimar.x.ro.       admin\.dns.x.ro. 1 3600 60 7200 60
                           NS     nsprimar.x.ro.
; pentru statia 10.0.0.1
1.0.0.10.in-addr.arpa.     PTR    c1.x.ro.
; pentru statia 10.0.0.200
200.0.0.10.in-addr.arpa.   PTR    c200.x.ro.
```

Nota: *a se observa faptul ca nu suntem obligati ca serverele autoritative ale unei zone sa aiba nume chiar din zona in cauza; putem folosi nume deja definite in alta zona.*

10.2.6. Verificarea, diagnosticarea si controlul functionarii serverului

10.2.6.1. Validarea fisierelor de configurare/zona

Pentru validare, suita BIND ne pune la dispozitie urmatoarele utilitare:

- **named-checkconf** – primeste ca argument optional calea catre fisierul de configurare; daca nu este specificat, foloseste locatia default a fisierului

```
named-checkconf                          # valideaza fisierul de configurare implicit
named-checkconf /etc/bind/named.conf     # valideaza fisierul solicitat
```

- **named-checkzone** – verifica validitatea fisierelor zona. Necesita doua argumente - numele zonei si calea catre fisierul zona:

```
named-checkzone x.ro /var/named/x.ro
```

Atentie! *In unele distributii, aceste utilitare se instaleaza separat de server; spre exemplu, in Ubuntu exista un pachet* bind9utils.

10.2.6.2. Rulare server in foreground

Una dintre principalele modalitati de diagnostic este pornirea serverului in foreground cu output-ul pe ecran, folosind optiunea **-g**:

```
root@server# named -g
```

In acest mod de functionare, serverul ofera o multitudine de informatii utile:

- fisierul de configurare folosit
- eventualele erori din fisierul de configurarea
- eventualele erori din fisierele zona, si daca acestea au fost incarcate cu succes. Spre exemplu, cand vedem mesajul *"zone x.ro/IN: loaded serial 15"* vom sti ca zona in cauza a fost considerata valida si incarcata cu succes

10.2.6.3. Interogare manuala

Odata ce serverul a pornit, a-i verifica corecta functionare sau a-l diagnostica inseamna, printre altele, a-i adresa aceleasi interogari pe care i le vor formula mai tarziu clientii sai si a ne asigura de corecitudinea raspunsurilor. In acest scop suita BIND ne ofera cateva utilitare de interogare manuala:

- **dig** – este utilitarul principal folosit pentru interogare DNS. Este cel mai bogat in facilitati si, in acelasi timp, output-ul sau se mapeaza pe formatul mesajului DNS. Acesta va fi cel prezentat in continuare
- **host** – este un utilitar mai simplu, gandit pentru obtinerea de raspunsuri rapide (ex: *host numeDNS* va afisa adresa corespunzatoare numelui, *host adresaIP* va afisa numele corespunzator adresei, daca exista)
- **nslookup** – utilitar interactiv, prezent si in Windows

Atentie! *In unele distributii, aceste utilitare se instaleaza separat de server; spre exemplu, in Ubuntu exista un pachet bind9utils.*

Utilitarul **dig** are urmatoarea sintaxa generala (argumentele incluse intre <...> sunt optionale):

```
dig    <@server_interogat>        numeDNS       <tipRR>
```

Daca serverul interogat nu este specificat, vor fi folosite serverele de sistem din */etc/resolv.conf*. Atunci cand tipul de RR lipseste, el se considera implicit A.

A interoga un server DNS inseamna sa-i furnizam un nume DNS si un tip de RR – practic, primele doua campuri ale inregistrarii in fisierul zona – si el sa ne raspunda cu inregistrarea completa. Iata cateva exemple de interogari:

```
# lista de servere autoritative ale domeniului yahoo.com
dig    @193.231.236.25     yahoo.com     NS
```

```
# adresa statiei cu numele www.infoacademy.net (cele doua comenzi sunt echivalente)
dig @193.230.161.3  www.infoacademy.net A
dig @193.230.161.3  www.infoacademy.net

# care este serverul primar pentru domeniul google.com? Vor fi folosite serverele din resolv.conf
dig    google.com    SOA
```

Output-ul lui *dig* urmareste indeaproape formatul mesajului DNS:

```
student@Desktop $ dig @ns1.yahoo.com yahoo.com

; <<>> DiG 9.9.5-3ubuntu0.5-Ubuntu <<>> @ns1.yahoo.com yahoo.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27207
;; flags: qr aa rd; QUERY: 1, ANSWER: 3, AUTHORITY: 6, ADDITIONAL: 11
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1272
;; QUESTION SECTION:
;yahoo.com.                IN     A

;; ANSWER SECTION:
yahoo.com.          1800   IN     A      98.138.253.109
yahoo.com.          1800   IN     A      206.190.36.45
yahoo.com.          1800   IN     A      98.139.183.24
...restul output-ului a fost omis...
```

Observatie: *formatul in care dig afiseaza informatia este de asa natura gandit incat poate fi scris direct intr-un fisier zona – toate liniile ce nu reprezinta inregistrari sunt comentate.*

Am evidentiat in output-ul lui *dig* flag-ul **aa** (authoritative answer): acesta ne indica faptul ca am primit raspuns chiar de la unul dintre serverele autoritative pentru zona *yahoo.com*. Atunci cand ne diagnosticam propriul server, acesta trebuie sa raspunda cu *aa* setat pentru toate zonele configurate in *named.conf*!

Utilitarul *dig* poate fi folosit si pentru rezolutie inversa; in loc sa cautam manual inregistrarea PTR aferenta, putem folosi optiunea **-x** urmata de adresa IP al carei nume DNS dorim sa-l aflam:

```
student@Desktop $ dig @8.8.8.8 -x 193.231.236.30
;; ANSWER SECTION:
30.236.231.193.in-addr.arpa. 21099 IN   PTR    dns-cache-2.rdsnet.ro.
```

10.2.6.4. Utilizarea lui rndc

Odata ce configurarea comunicatiei rndc-server a fost efectuata corect, putem folosi utilitarul in cauza pentru a controla serverul, dupa cum urmeaza:

- **rndc reload** – cere serverului sa reciteasca din mers fisierul de configurare. Poate primi un alt doilea argument ce reprezinta un nume de zona, pentru a reincarca doar informatia zonei respective; exemplu: *rndc reload a.ro*
- **rndc stop** – opreste serverul
- **rndc status** – afiseaza un scurt sumar al starii serverului (versiune, numar de zone gazduite, numar de clienti serviti in momentul respectiv, numar de transferuri de zona in desfasurare etc)

- **rndc notify *nume_zona*** – trimite mesaje DNS de tip *notify*, care forteaza serverele secundare ale zonei specificate sa se actualizeze, efectuand transfer de zona daca este cazul
- **rndc retransfer *nume_zona*** – in cazul unui server secundar, forteaza transferarea zonei de pe master
- **rndc flush** – goleste cache-ul serverului
- **rndc dumpdb** - salveaza continutul cache-ului (care in mod normal sta numai in RAM) in fisierul specificat cu directiva *dump-file* in fisierul de configurare
- **rndc querylog** - activeaza sau dezactiveaza consemnarea in loguri a interogarilor primite de la clienti. Util in conjunctie cu rularea serverului in foreground

10.3. BIBLIOGRAFIE

- Standardul DNS (RFC 1034+RFC 1035): http://www.rfc-editor.org/rfc/std/std13.txt
- Domenii internationalizate: https://en.wikipedia.org/wiki/Internationalized_country_code_top-level_domain
- BIND Administrator's Reference Manual: http://ftp.isc.org/isc/bind9/9.10.2/doc/arm/Bv9ARM.html
- Cartea *DNS and BIND* - editura O'Reilly