Az Internet DNS – Elv és konfiguráció

Pásztor Miklós – MTA SZTAKI/ASZI (*pasztor@sztaki.hu*)
Lektorálta: Kiss Gábor, Martos Balázs

Az Interneten használt osztott név adatbázis, a **DNS** (**D**omain **N**ame **S**ervice) folyton használatos: minden weblap letöltésnél, levél közvetítésnél szerepe van, nélküle megbénulna a hálózat, mégis sokan még a létezéséről sem vesznek tudomást, a szolgáltatás csendesen dolgozik a háttérben.

A DNS egy **osztott, hierarchikus adatbázis.** Az adatbázist jelenleg név szerverek százezrei szolgáltatják nevek millióiról. A tervezéskor gondoltak a redundanciára és a hibatűrésre: a névszerverek sokszor nem érhetők el, konfigurációjuk tele van hibával, hiányossággal, elavult adatokkal, az egész mégis bámulatos módon működik. A DNS rendszer legfontosabb feladata a név–IP cím feloldás, de – ahogy azt látni fogjuk –, egy sor más információt is szolgáltat a domain nevekről. A rendszergazdák fontos feladata a DNS konfigurálás. Ebben a írásban a DNS rendszer – nem is bonyolult – elvét ismertetjük, és leírjuk a konfigurálás legfontosabb elemeit.

IP címek, nevek

Az Internethez kapcsolódó hálózati eszközök, számítógépek mindegyikének egyedi azonosítója, (32 biten tárolt) IP címe van. A felhasználók azonban olyan neveket szeretnek használni, amelyek könnyebben megjegyezhetők, mint egy ilyen hosszú szám, és a névből következtetni tudnak a gép, a szolgáltatás helyére, a szolgáltatás típusára is. Ezért kezdettől fogva neveket rendeltek az IP címekhez. Amikor az Internet még csak pár ezer számítógépből állt, ezt a név-cím hozzárendelést egy folyamatosan növekvő állomány, **host táblázat** tartalmazta. Ezt a táblázatot minden számítógépen lokálisan tárolták, és egy központi helyről rendszeresen frissítették. Ennek nyoma mind a mai napig megvan: pl. a unix rendszerekben az `/etc/hosts` fájl éppen ilyen.

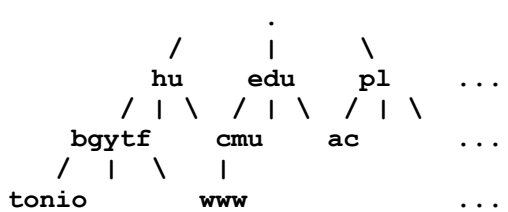
Az Internet növekedtével azonban ez a megoldás tarthatatlanná vált: a fájl hatalmasra dagadt, egyre sűrűbben kellett módosítani, egyre többen töltötték le, egyre gyakrabban, Ezért jött létre a DNS (Domain Name Service), az internetes kommunikáció egyik fundamentuma. Kidolgozásában fő szerepet játszott P. Mockapetris, az ISI (Information Science Institute) munkatársa. A DNS elve egyszerű és ötletes, frappáns bizonyítéka annak, hogy a szubszidiaritás elve milyen jól működik a gyakorlatban.

A nevek feloldása hálózati kommunikáció által történik. A névszerverek feladata kettős:

- **látni –** azaz az elosztott DNS adatbázist kérdezni, a hálózati szolgáltatások számára az érvényben levő név-cím hozzárendelésről információt adni és
- **láttatni,** mutatni – az elosztott adatbázis ide kiosztott részére információforrásként viselkedni, azaz a nevek egy bizonyos halmazáról a többi név szerver számára – mint illetékes – adatokat szolgáltatni.

Ha egy név dolgában egy szerver az Internet számára elsődleges információforrás, azaz illetékes, azt úgy szokás kifejezni, hogy az ő adata **autoritatív.**

Mindenki ismer internet neveket: mail.whitehouse.com, reklam.radio.hu. Az internet nevek fordított fa szerint szerveződő hierarchiát alkotnak:



A fa **fordított**, mert a gyökér a hierarchia legmagasabb foka. A nevek feloldása a gyökértől kezdődik, és fokról fokra halad előre. A név-fa különböző elágazási pontjaiért és ágaiért különböző szerverek felelősek. Egy-egy szerver több ágért is felelős lehet. A név-fa egy-egy pontját **domain**-nak, **domain névnek** vagy egyszerűen **név-**nek nevezzük.

A név hierarchia

A hierarchia csúcsát „root”-nak, gyökérnek nevezzük. Az ez alatti neveket **top level domain**-oknak, TLD-knek mondjuk. Amikor az Internet még csak USA hálózat volt, a következő TLD-k voltak használatosak:

edu – amerikai egyetemek, oktatási intézmények
com – vállalatok
mil – katonai szervezetek
gov – kormányhivatalok
net – hálózati szervezetek
org – mindenféle más szervezet
arpa – az Internet ősében, az Arpanetben levő gépek neveire szolgált kezdetben. Az inverz nevek feloldásánál (ld. később) mind a mai napig fontos szerepe van.

Az USA-n kívüli domain-ok számára az ISO 3166 szabványban meghatározott kétkarakteres országkódot kezdték használni. Példák:

be – Belgium
pl – Lengyelország
hu – Magyarország

A hierarchia nagyon hasonlít az operációs rendszerek hierarchikus fájlstruktúrájához (pl. `C:\anyagok\majus\jelentes1.txt`), csak az alá-fölérendeltség itt éppen fordítva, jobbról balra olvasható le. Pl. gep.csoport.osztaly.intezet.hu. A TLD elnevezés mellett használatos még az SLD (second level domain) kifejezés is, a hierarchia második szintjén levő domain-okra.

Zónák

A név-fa zónákra oszlik: egy-egy zóna a **fa egyben kezelt része.** Sokszor – de nem feltétlenül, – egybeesik egy aldomain-nel. Például egy zóna lehet az osztaly.intezet.hu és minden név, ami a hierarchiában ez alatt van. Egy zóna például az összes TLD-t tartalmazó root zóna is. Egy zóna a „láttató”, az „autoritatív” szerver szempontjából egy egység, rendszerint egy fájl. Egy-egy zónát több szerver is láttat(hat). Ezek közül az egyik az elsődleges, a többi (ha van) másodlagos.

Az elsődleges szerveren az adatok a zóna adminisztrátor munkájának eredményeképpen ténylegesen változnak.

A másodlagos szerver(ek) a zóna adatait meghatározott rend szerint az elsődleges szervertől tükrözi(k). A tükrözés rendjét az elsődleges szerveren a rendszeradminisztrátor a zóna konfigurációjával határozza meg.

Delegálás

A hierarchia egyes darabjait a zóna adminisztrátora tovább delegálhatja más szerverekre. Például az intezet.hu domain gazdája az osztaly.intezet.hu aldomain láttatását, autoritását az illető osztály egy meghatározott gépére bízhatja a konfigurációban: mindenki felelős és úr lehet a saját illetékességi körében (szubszidiaritás elve). A root zóna sőt még a TLD-k (edu, gov, hu stb.) is jóformán mást sem tartalmaznak mint ilyen delegálást. Így jön létre a hierarchikus, osztott adatbázis. A delegálás azonban nem feltétele a többszintű név megadásának. Például lehetséges, hogy az osztaly.intezet.hu nincs delegálva, nem különálló zóna, mégis létezik a gep.osztaly.intezet.hu domain, mert az

intezet.hu zóna gazdája bevezette a pontot (.) tartalmazó gep.osztaly nevet. Ezt éppen úgy megteheti, mint a gep-osztaly vagy az osztalygepe nevek bevezetését, melyeknek hatása a gep-osztaly.intezet.hu, illetve az osztalygepe.intezet.hu nevek létrejötte.

Domain nevek

A hierarchia következtében **minden név egyedi.** Lehet, hogy az Internet több pontján is elneveznek egy gépet pl. jupiter-nek, de nevük egyértelmű, ha a teljes domain nevüket mondjuk:

```
jupiter.osztaly.intezet.hu.
jupiter.arizona.edu.
```

A domain neveknek ezt a teljes alakját, ami a nevet a gyökér domain-ig tartalmazza **FQDN**-nek (Fully Qualified Domain Name), a domain név pontokkal elválasztott darabjait pedig **szegmenseknek** nevezzük. Annak jelzésére, hogy a domain név teljes, a név végére pontot teszünk. Valójában a TLD-re (hu, edu) való végződés nem garantálja, hogy a név FQDN: elképzelhető és tökéletesen szabályos a jupiter.arizona.edu.osztaly.intezet.hu domain név is.

Domain nevekben megengedett karakterek a **latin ABC** betűi [a-z], a **számjegyek** [0-9] és a **kötőjel** (-). Kis- és nagybetű egyformán használható, és nem jelent különbséget. Sajnos nem állhat domain névben ékezetes karakter. **[A leírás keletkezése óta ez megváltozott, ma már ékezetes karakterek is felhasználhatók!]** Gyakori hiba, hogy aláhúzás (_) karaktert adnak meg domain nevekben. Az eredeti definició (RFC1035) az egyes szegmensek elején csak betűt engedett meg, a későbbi (RFC1123) megengedi a számmal kezdődő szegmenst is. Például szabályos a 3com.com domain. Kötőjel viszont nem állhat továbbra sem se szegmens név elején, sem végén.

Cím → név hozzárendelés

Az Interneten nem csak arra van szükség, hogy nevekből IP címeket nyerjünk, hanem arra is, hogy **IP címekből domain neveket.** Ez a szolgáltatás – amit inverz, vagy reverz feloldásnak neveznek –, a hálózati biztonság szempontjainak erősödése miatt egyre nagyobb jelentőségű. Például sok FTP vagy levelező szerver nem fogad el kéréseket csak olyan gépekről, amiknek címéből a hozzájuk tartozó domain nevet ki lehet deríteni. Vannak szolgáltatások, amik csak bizonyos domain-okból érhetők el.

A cím-név feloldás érdekében bevezették az in-addr.arpa domaint. IP címeket általában úgynevezett pontozott decimális (dotted decimal) alakban szokás megadni, ilyesformán: 150.151.152.153. Az ehhez a címhez tartozó nevet úgy kapjuk meg, hogy a domain rendszertől megkérdezzük a 153.152.151.150.in-addr.arpa névhez tartozó rekordot.

Az in-addr.arpa domainban éppen úgy delegálják az egyes aldomain-eket mint minden más zónában.

Rezolverek és DNS szerverek

Hogyan is zajlik a névfeloldás? Tételezzük fel, hogy a jupiter.arizona.edu nevet kell feloldani, mert pl. oda akarunk egy levelet továbbítani, vagy ftp-vel belépni. Ezért az általunk használt programnak – pl. a web böngészőnek –, megadjuk a jupiter.arizona.edu domain nevet. Programunknak ekkor meg kell állapítania, hogy milyen IP cím is tartozik ehhez a domain névhez. Ezt a funkciót ellátó egységet nevezzük **rezolvernek, feloldónak.** Gépünkön a TCP/IP szoftver telepítésekor, konfigurálásakor meg kellett adni egy vagy több DNS szervert. Ezekhez fordul a rezolver. A DNS szerver lehet a gép saját maga, vagy – elvben – tetszőleges gép az Interneten. Tehát elvben lehetséges, hogy egy Indonéziában levő számítógép egy Dániában levő name szervert állít be a rezolver konfigurációjában. Persze ez ésszerűtlen. Célszerű egy hálózati értelemben közeli szervert kijelölni. A rendszergazdák kedves kötelessége erre vonatkozó információval ellátni felhasználóikat. A rezolver rendszerint néhány konfigurációs fájlból és könyvtári szubrutinból áll.

Gyakorlatilag minden TCP/IP-t használó, Internetbe kapcsolt számítógépen szükség van rá. A rezolver tehát nem végez közvetlenül névfeloldást, hanem bizonyos általa ismert névszervereket kér meg arra, hogy a feloldást elvégezzék.

A rezolver konfigurációban a DNS szerverek megadásánál értelemszerűen IP címeket kell használnunk. Sok konfiguráló program a szerverek megadásánál használja az „elsődleges" (primary), „másodlagos" (secondary) kifejezéseket. Ez gyakran zavart okoz, mert összekeverik a zónáknál használatos hasonló kifejezésekkel. A **rezolver** konfigurációnál megadott elsődleges/másodlagos névszerver a **látásra** vonatkozik, vagyis arra, hogy kliensünk milyen név szervereket kérdez. A zóna definíciónál pedig az elsődleges névszerver az, amiről a másodlagos szerverek **tükrözik** a láttatott, mutatott zónát.

Amikor a rezolver a konfigurációjában megadott névszerverhez fordul, hogy például a jupiter.arizona.edu névhez tartozó IP címet megtudja, akkor a szerver általában nem válaszol azonnal. Példánkban legyen a kérdezett névszerver a ns.intezet.hu. Az ns konfigurációjának archimédeszi pontja – hasonlóan a rezolver konfiguráció DNS szerver IP címeihez –, a **gyökér névszerverek IP címe**. Ezek valamelyikét kérdezi az ns névszerver. Egy root névszervert kérdezve például a jupiter.arizona.edu névről, az nem ad mást, mint a .edu zónáért felelős név szerverek listáját. Az ns névszerver ekkor egy újabb kérdést intéz a .edu névszerveréhez, aki újra csak arra vonatkozóan ad információt, hogy hova lehet fordulni az arizona.edu nevek feloldásáért. Ilyen módon a ns rekurzív módon oldja fel a nevet, melynek végén a kérdező kliens gép rezolverének megadja a választ. A DNS szerverek általában nem végeznek bármely kliens számára ilyen rekurzív feloldást, hanem csak a konfigurációjukban meghatározottakra.

Cache, TTL

A névszerverek az általuk megtudott neveket tárolják azzal a céllal, hogy ha újra megkérdezik tőlük, akkor ebből a cache-ből **azonnal tudjanak válaszolni.** Ennek többszörös haszna van: **csökkenti** a hálózati forgalmat, és **gyorsítja** a névfeloldást. A cache-ben minden megtudott nevet, csak egy bizonyos ideig tárolnak. Ha ez az idő lejárt, akkor egy újabb kéréskor – hiába lenne a cache-ben az információ-, a névszerver újra kérdezi azt. Ilyen módon, ha a névhez tartozó információ esetleg változik, arról tudomást szerezhet. Azt az időt, ameddig a cache-ben van egy-egy információ, nem a tárolóban, hanem a láttató, az autoritatív szerverben döntik el: minden rekordhoz tartozik egy – sokszor implicit módon megadott – **TTL (Time To Live)** érték. Ennyi másodpercig tárolják a szerverek a cache-ükben az információt.

Névszerverek funkció szerint

Caching only szerverek

A névszerverek egy része nem autoritás semmilyen névre, hanem csak arra szolgál, hogy feloldja a neveket a kliensek számára. Ezeket nevezzük „caching only" – csak cache-elő – névszervereknek. Általában ajánlatos minden lokális hálózaton legalább egy névszervert működtetni. Ha nincs „láttató" feladat, akkor caching-only szerverre van szükség.

Láttató, autoritatív szerverek

Ahogy már erről szó volt, ezek azok a név szerverek, melyeknek az (is) feladata, hogy bizonyos neveket ők mutassanak meg mások számára. A domain név-fa egy **egyben delegált ágát, melyért egy szerver felelős, zónának nevezzük.** Egy zónáért felelős névszerverek közt van egy kitüntetett, amelyen az **adminisztrátor** a konfigurációt változtatja. Az (esetleges) többi ezt a zónát tükrözi. A kitüntetett szerverre elterjedt kifejezés az „elsődleges", „primary" a tükröző szerverekre pedig a „másodlagos", „secondary". Újabban (elsősorban a 8. változatú BIND megjelenésének hatására) inkább a **master** és a **slave** neveket használják. A master és slave név azért szerencsésebb, mert nem keveredik a rezolver konfigurációknál megadható „primary/secondary" szerverekkel. Sajnos a

„slave” szerver kifejezés is használatos már régebben és más értelemben: az olyan szerverekre mondjuk hogy „slave”, amelyik csak forwarderek közvetítésével érintkezik az Internet nagyobb részével. Egy szerver lehet egy zónára „master” egy másikra „slave”. Valójában gyakori is, hogy két intézmény kölcsönösen „slave” autoritatív szerver a egymás zónáira. A névfeloldás szempontjából a „master” és a „slave” szerverek között semmi különbség nincsen: egyformán autoritatív mindegyik. A névszerverek a név feloldás során bármelyikhez fordulhatnak. A valóságban a kód úgy működik, hogy a szerverek egy-egy zóna autoritatív szerverei közül igyekeznek azt kérdezni, amelyik gyorsabban válaszol, aminek érdekében egy ravasz algoritmust használnak: kezdetben mindegyik névszervert megkérdezik, mérik a válaszidőt, aztán azt preferálják, ami gyorsabban válaszolt, de a lassabb szerverek idővel újra szót kaphatnak, mert minden kérdésnél „csökken a büntetésük”.

Forwarder szerverek

Egy névszerver gyakorlatilag kiegészítheti a cache-ét más szerverek cache-évél, ha a forwarder opciót használják a konfigurálásánál. Ha pl. kicsi.valahol.hu gépen a DNS konfigurációban megadják, hogy a nagy.valahol.hu forwarder legyen számára, akkor a kicsi-n történő névfeloldás úgy zajlik, hogy ha a kicsi cache-ében nincs benne a kért név, akkor a kicsi DNS szerver mielőtt a világban a név-fa hierarchiának megfelelő módon elkezdene érdeklődni , megkérdezi a nagy-ot. Ha annak a cache-ében megtalálható a keresett rekord, akkor válaszol, és így a kicsi gyorsan megtalálja a választ. Elképzelhető, hogy egy-egy intézménynél több kisebb szerver használ egy közös nagyobb forgalmú forwardert. Például nem csak a kicsi.valahol.hu, hanem a pici.valahol.hu is a nagy.valahol.hu-t. A több irányból érkező, több kérés hatására a nagy.valahol.hu-nak *nagy* cache-e keletkezik.

Slave szerverek

Az olyan szervert, ami csak forwardert (esetleg többet) használ a nevek feloldására, slave szervernek nevezzük. Slave szerverre van szükség tűzfal mögött, ahol a szervernek módja sincs, hogy közvetlenül kilásson az Internetre. Ahogy már említettük ez a fajta „slave” fogalom nem keverendő össze a „slave” fogalmával egy-egy zóna szempontjából: a forwarder(ek)re támaszkodó slave szerver korlátozott a **látás** szempontjából, egy-egy zóna slave szervere pedig az illető zóna **mutatása, láttatása** szempontjából.

Zónafájlok

A névszerverek az egyes zónák adatait általában egy-egy fájlban tárolják. A „master” szerveren az adminisztrátor személy közvetlenül, vagy valamilyen program közvetítésével maga módosítja ezt a fájlt. A „slave” szervereken a fájl a tükrözés eredménye.

A zónafájl rekordokból, RR-ekből (resource record) áll. Nagyon sok fajta rekordot tesznek lehetővé az RFC-kben megadott definíciók. A következőkben ezek közül ismertetjük a legfontosabbakat.

A rekordok formáját az RFC1035 határozza meg, és az a következő:

```
cimke ttl osztály típus adatok
```

A „**cimke”** a domain rekord neve. Lehet üres, ilyenkor az előtte levő rekord címkéje érvényes. A „**ttl**” a rekordhoz tartozó time to live időt adja meg másodpercben. Nem kötelező paraméter. Ha elhagyjuk, akkor a zónára vonatkozó alapértelmezés lesz a rekordhoz tartozó érték. A következő paraméter értéke gyakorlatilag mindig **IN,** azaz internet osztály. Ez is elhagyható. A **„típus”** mondja meg, hogy milyen fajta információról is van szó. Pl. IP cím (A rekord), name szerver információ (NS rekord) stb. Az **„adatok”** mező a rekord típusától függő információt tartalmaz.

Rekordok

SOA – Start of Authority rekord, zóna kezdő rekord

A SOA rekord adja meg egy zónára vonatkozó közös információkat. A rekord formáját egy példán mutatjuk be:

```
valami.hu.   SOA       gep.valami.hu.       mester.valami.hu. (
                       1999093001           ;Serial nr.
                       86400                ;Refresh
                       1800                 ;Retry
                       604800               ;Expire
                       43200)               ;TTL
```

A **cimke** (valami.hu.) a zóna neve. A SOA kulcsszó utáni első paraméter a zónához tartozó elsődleges szerver domain neve. A második paraméter egy e-mail cím, melyet úgy kapunk, ha az első olyan . karaktert, amit nem előz meg backslash (\) , at jelre, @-re cseréljük. A **serial nr.** a zóna sorszáma. Arra szolgál, hogy a slave (másodlagos) szerverek ellenőrizhessék, hogy a náluk levő zóna tartalom nem avult-e el. Akkor töltik le az master (elsődleges) szerverről a zóna tartalmát, ha a náluk levő zóna sorszám kisebb. Arra kell tehát vigyázni az elsődleges szerver adminisztrátorának, hogy ez a szám mindig növekedjen, ha valamit változtat, ha új változat keletkezik a zónából. Szokás ezt a sorszámot ÉÉÉÉHHNNVV alakban megadni, ahol ÉÉÉÉ az év négy jegyen, HH a hónap két jegyen, NN a nap két jegyen, VV a napon belüli változat két jegyen ábrázolva. Az ez után következő négy paraméter mind **másodpercben** megadott érték. Az első a **refresh, a frissítés idő** azt mondja meg, hogy mennyi időnként kell a slave szervereknek a master-től megkérdezni, hogy a zóna sorszáma mennyi, vagyis, hogy szükséges-e a zónát frissíteni náluk. A **retry** idő azt mutatja, hogy ha a frissítés nem sikerült, akkor mennyi időt várjanak, mielőtt újra próbálkoznának. Az **expire** azt mondja meg, hogy ha nem sikerül a master-rel kommunikálniuk, ennyi ideig szolgáltatják a zónát a világ számára. A TTL érték lesz a zóna rekordjaira érvényes alapértelmezés.

Figyelni kell rá, hogy észszerűen állítsuk be a zóna SOA rekordjában az idő értékeket. A legtöbb esetben az 1 napos (86400) refresh, 1 órás (3600) retry, 1 hetes (604800) expire és 1 napos (86400) TTL megfelelő. Ha gyors változás várható, akkor érdemes a TTL értéket kicsire venni. A dolog természetéből adódóan súlyos zavarokat okoz, ha az expire idő nem nagyobb mint a refresh: a másodlagos zóna nem fogja szolgáltatni az adatokat az idő egy részében.

A 8-as változatú Bind-nál a másodpercben értendő dimenzió nélkül megadott számok helyett használhatunk emberek számára könnyebben kezelhető mértékegységekben megadott számokat, ilyenformán:

```
1W2D3H
```

A W (week) heteket, D (day) napokat, H (hour) órákat jelent.

A – Address, cím rekord

Ez a leggyakrabban használt rekord, amely arra szolgál, hogy egy domain névhez IP címet rendeljünk. Például:

```
masina A 190.111.222.3
```

Sokszor használt tulajdonságát látjuk itt a zónafájlnak: nem írjuk ki egy domain (jelen esetben a masina) teljes domain nevét, csak annak első részét. A végére oda kell érteni azt a **vonatkoztatási rendszert,** ahol éppen vagyunk. Ezt először is maga az a zóna adja meg, amire ez a fájl vonatkozik. Például ha a valami.hu zónáról van szó, akkor a „masina” a végére biggyesztett pont nélkül úgy értendő, mint masina.valami.hu. Ez a tulajdonság legtöbbször igen kellemes, mert például egy 200 A rekordot tartalmazó zóna esetében nem kell 200-szor megismételnünk a zónában, hogy „egyik.valami.hu., masik.valami.hu.” hanem elég annyit írnunk „egyik, masik”. Vigyáznunk kell azonban, mert könnyen elfelejtkezhetünk arról, hogy pontot kell tennünk a domain név végére, ha

azt teljes egészében kiírjuk valahol. Figyeljük meg ebből a szempontból a SOA rekordra felhozott példát fentebb.

NS – Name Server, névszerver rekord

Ez a rekord szolgál arra, hogy egy domain névszervereit megadjuk. Ilyen módon a domain egy delegálási pont. Példa:

```
osztaly NS gep.osztaly.valami.hu.
```

Ezzel a rekorddal deklaráljuk, hogy az „osztaly” aldomain névszervere a gep.osztaly.valami.hu. Ajánlatos – bár technikai értelemben nem kötelező – legalább két névszervert megadni. Ilyen módon a zóna adatai akkor is elérhetők a világból, ha az egyik gép, vagy a hozzá vezető vonal valami miatt kiesne. A felsőbb szinten – példánkban a valami.hu zóna alatt –, nem látszik, hogy a szerverek közül melyik a master és melyik a slave. Szigorúan véve az NS rekordoknak csak a felsőbb szinten, az „apuka” zónában van szerepe, indokolt azonban a zónában is felsorolni. Az NS rekord paramétere egy gép domain neve. Szükséges, hogy ehhez a névhez közvetlenül A rekord tartozzon. Elő-előfordul, de hibás CNAME rekorddal definiált domain nevet megadni.

Glue rekord

Gyakori, hogy a delegált zóna egyik name szervere éppen a zónában van, mint a fenti példában. A gep.osztaly.valami.hu rekordnak az osztaly zónában van a helye, de mégis szükség van arra, hogy egy szinttel feljebb, a valami.hu zónában is felsoroljuk, különben csapdába kerülünk. Ezért fel kell vennünk egy nem oda való A rekordot:

```
gep.osztaly A 190.1.2.3
```

Az ilyen, idegen A rekordot nevezik glue (ragadvány) rekordnak. Előfordul, hogy adminisztrátorok akkor is felsorolnak nem a zónába való A rekordot, amikor az nem egy onnan delegált aldomainban van, ez hiba. Semmi haszna és zavart okoz. Tehát például ha az osztaly.valami.hu zónának egy másik névszervere a mas.nevszerver.intezet.hu, akkor ehhez nem kell glue rekordot csatolni a valami.hu zónában, hiszen ennek a névszervernek az A rekordját ettől a delegálástól teljesen függetlenül lehet megtudni.

Lame delegálás

Ha valahova delegálunk egy zónát, akkor az ottani adminisztrátorral meg kell beszélnünk, hogy azt folyamatosan szolgáltassa is. Ha ez nem történik meg, akkor beszélünk „lame” delegálásról. Sokszor előfordul például amiatt, mert a delegált zóna slave szervere nevet változtat, vagy meg is szűnik, és erről elfelejtik értesíteni a felettes zóna gazdáit.

CNAME – Canonical Name, kanonikus név rekord

Ez a rekord arra való, hogy egy hostnak becenevet adjunk. Például:

```
www CNAME gep
```

Ha ez a rekord van mondjuk a valahol.hu zónában, az azt mutatja, hogy a www.valahol.hu egy másik neve a gep.valahol.hu-nak. Nagyon hasznos az ilyen név például a következő esetben: tegyük fel, hogy egy idő után a gep.valahol.hu meg is szűnik, és a szolgáltatást az ujdivat.valahol.hu veszi át. Ilyenkor elég csak a CNAME rekordot módosítani, így:

```
www CNAME ujdivat
```

A világ számára a valahol.hu weblapjai továbbra is a www.valahol.hu gépen lesznek elérhetők. Az is gyakori, hogy egy gép több funkciót is ellát, és a funkciók mindegyikéhez tartozik egy-egy CNAME rekord, ami ugyanarra a gépre mutat. Például news.valahol.hu, ftp.valahol.hu mind mutathatnak ugyanoda.

Mint látjuk, a CNAME rekord paramétere egy domain név. Általában ez a név már A rekorddá oldható fel. Megengedett, de nem ajánlatos a CNAME-ra mutató CNAME rekord.

MX – Mail eXchanger, levelező szerver rekord

Ez a rekord szolgál arra, hogy egy domainba érkező levelek levelező szerverét kijelölje. A rekord formátuma egy példán:

```
valahol.hu.   MX      10      masina.valahol.hu.
              MX      20      mas.mashol.hu.
```

Ezek a sorok azt jelentik, hogy a valaki@valahol.hu alakú címre érkező leveleket a masina.valahol.hu, vagy a mas.mashol.hu. gépekre kell küldeni. Az MX rekordok első paramétere egy szám, ami a rekord preferenciát jelenti. Kötelező paraméter, de csak akkor van jelentősége, ha több MX rekord tartozik ugyanahhoz a névhez: kisebb szám nagyobb preferenciát jelent. Példánkban tehát csak akkor fogják a levelező szerverek a mas.mashol.hu-ra küldeni a valahol.hu domainba szóló leveleket, ha a preferáltabb masina.valahol.hu nem elérhető. Lehetséges több MX rekordot egyenlő preferenciával megadni. Ilyenkor véletlenszerű, hogy melyikre érkezik be egy-egy levél. Az MX rekord második paramétere egy domain név. Fontos, hogy ehhez a névhez már A rekord tartozzon. Nem megengedett olyan domain nevet megadni, ami csak egy CNAME-re, vagy másik MX-re mutat.

Az MX rekord gyakori alkalmazása, amikor egy intézményben egységes, egyszerűsített, és könnyen megjegyezhető levélcímeket vezetnek be a segítségével. Például a Firma cégnél kovacs.janos@firma.hu alakú levelezési címe lehet mindenkinek, ha a firma.hu MX rekord egy – akár időben változó – levelező szerverre mutat, ahol aztán feloldják a levél cím első részében a név aliast, esetleg tovább küldik a levelet egy másik szerverre.

TXT – Text, szöveges rekord

Ez a rekord tetszőleges szöveges információt tartalmazhat. Példa:

```
modern      TXT     "Ez a gep mar megszunt"
```

A TXT rekord paramétere egyetlen, idézőjelek közé zárt ASCII karaktersorozat.

HINFO – Hardware information, hardver információ rekord

Akárcsak a TXT rekord ez a rekord is emberi olvasásra szánt, egy számítógépről nyújt felvilágosítást. Példa:

```
masina      HINFO VAX     VMS-4.7
```

Mint látható, két paramétere van. Az első a hardver típust, a második az operációs rendszert szokta jelölni.

PTR – Pointer rekord

Ahogy arról már szó volt, nem csak név-cím, hanem cím-név hozzárendelésre is szükség van. Ezt a szolgáltatást elsősorban nem emberek, nem is kliens programok, hanem szerver programok használják, annak kiderítésére, hogy egy hozzájuk érkezett IP csomag milyen domainhoz is tartozik. DNS rendszerben az in-addr.arpa domain alá tartozó ág szolgálja a cím-név felosztást. Itt a zónák delegálása az IP címtartomány egyes darabjainak megfelelően történik. Példa:

```
140.in-addr.arpa. NS     ...
```

Ez a zóna a 140.x.y.z alakú IP címek inverz domain név szolgáltatásánál játszik szerepet. Ha egy intézmény egy C osztályú címet kap, vagyis gazdálkodhat pl. a 192.84.124.x alakú címekkel, akkor célszerű, ha nála van a 124.84.192.in-addr.arpa zóna elsődleges névszervere is. Ebben a zónában vannak azután a PTR rekordok. Például:

```
22    PTR   gep.valahol.hu.
```

A PTR rekord egyetlen paramétere az a domain név, ami az illető IP címhez tartozik. A paraméterként megadott domain név A rekorddá kell forduljon az „egyenes” feloldáskor.

Amikor egy domain-adminisztrátor – elterjedt kifejezéssel „hostmaster” – egy gépnek, vagy valamilyen hálózati interfésznek nevet, és IP címet oszt, fontos, hogy gondoskodjon az **inverz feloldásról is**: általában párhuzamosan van szükség egy-egy A rekord és PTR rekord bejegyzésére. A dolog természete miatt az „egyenes” és az inverz zónák **nem járnak feltétlenül együtt.** Ha az osztaly.intezet.hu zónát kezeljük, és gazdálkodunk egy IP címtartománnyal, akkor nem nyilvánvaló, hogy mi is a kiosztott IP címekhez tartozó inverz zóna, vagy hogy azt egyáltalán mi kezeljük. Kezdő domain név adminisztrátoroknál gyakori hiba, hogy elfelejtkeznek az inverz domainról. A DNS hierarchikus szerkezetéből következik azonban, hogy bárki számára egyértelműen kideríthető, hogy ki is a felelős az általunk osztott IP címekhez tartozó in-addr.arpa zónáért. Neki kell azután szólni, hogy a megfelelő bejegyzést végezze el, vagy delegálja tovább a zóna egy darabját nekünk. Például ha a „host” parancsot használjuk a DNS nézegetésre, és arra vagyunk kíváncsiak, hogy kinek is kell bevezetni a 202.103.132.169 IP címhez tartozó inverz rekordot, akkor a következő láncon haladhatunk:

```
%host -t any 202.in-addr.arpa
202.in-addr.arpa        NS    NS.RIPE.NET
202.in-addr.arpa        NS    NS.TELSTRA.NET
202.in-addr.arpa        NS    NS.APNIC.NET
202.in-addr.arpa        NS    SVC00.APNIC.NET
202.in-addr.arpa        SOA   NS.APNIC.NET inaddr.APNIC.NET (
                        1999091501        ;serial (version)
                        86400             ;refresh period (1 day)
                        7200              ;retry interval (2 hours)
                        2592000           ;expire time (4 weeks, 2 days)
                        345600            ;defaultttl (4 days)
                        )
```

Tehát a 202.x.y.z alakú IP címekhez tartozó inverz zónákat az ns.apnic.net gépen kezelik, és szolgáltatja még három másik name szerver.

Haladjunk tovább:

```
%host -t any 103.202.in-addr.arpa
103.202.in-addr.arpa    NS    ns.telstra.net
103.202.in-addr.arpa    NS    svc00.apnic.net
103.202.in-addr.arpa    SOA   ns.apnic.net    inaddr.apnic.net (
                        1999081001        ;serial(version)
                        86400             ;refresh period (1 day)
                        7200              ;retry interval (2 hours)
                        2592000           ;expire time (4 weeks, 2 days)
                        345600            ;default ttl (4 days)
                        )
```

A 202.103.x.y alakú IP címek zónájának hazája ezek szerint szintén az ns.apnic.net.

És itt:

```
%host  -t any 132.103.202.in-addr.arpa
132.103.202.in-addr.arpa does not exist, try again
```

és:

```
%host -t any 169.132.103.202.in-addr.arpa
169.132.103.202.in-addr.arpa does not exist, try again
```

Vagyis a helyzet kulcsa annak a személynek a kezében van, akit az inaddr@apnic.net címen érhetünk el: vagy tovább kell delegálnia megfelelő helyre a 132.103.202.in-addr.arpa zónát, vagy neki kell bevezetnie a 169-es IP címhez tartozó PTR rekordot.

De Groot féle inverz feloldás

A klasszikus Interneten az IP címeket A, B, és C osztályú hálózati darabokban osztották, és amikor egy intézmény egy címtartományt kapott, pontosan meg lehetett mondani, hogy melyik a.in-addr.arpa, b1.b2.in-addr.arpa vagy c1.c2.c3.in-addr.arpa zóna tartozik a kapott címtartományhoz. Ennek a delegálását kellett az intézmény adminisztrátorának kérnie, és ettől kezdve könnyen kezelhette az egyenes és in-addr.arpa zónáit. A CIDR (Classless Inter-Domain Routing) elterjedésével gyakori, hogy egy-egy intézmény például csak egy negyed részét kapja meg egy C osztályú címnek. Az ilyen címtartományt úgy szokás jelölni, hogy a legkisebb használható cím után / jellel elválasztva megadjuk a tartományt jellemző bitmaszk egyeseinek számát. Például: 193.225.86.128/26 jelenti a 193.225.86.128-tól 193.225.86.191-ig terjedő címtartományt. Előfordulhat ilyen módon, hogy egy C osztályú cím 4 vagy még több egymástól távol eső intézmény között oszlik meg. Ha az ilyen tartományhoz tartozó inverz domaint a hagyományos módon szeretnénk kezelni, akkor minden intézménynek egy központi helyen kellene a PTR rekordjait beíratni. Ez bonyolult és kellemetlen. Sokkal jobb, ha – mint a klasszikus esetben – minden intézmény saját maga jegyezheti be a saját PTR rekordjait. A problémára Geert Jan de Groot adott megoldást, és azt az RFC2317 írja le. A megoldás a DNS technika szellemes alkalmazását mutatja.

A darabokra szabdalt C osztályú címhez tartozó in-addr.arpa zónában nem vezetünk be PTR rekordokat, viszont minden egyes rekordhoz bevezetünk egy CNAME rekordot. Ez a CNAME rekord olyan domain névre mutat, ami a címet birtokló intézmény adminisztrátora definiál. Például ha a 193.225.86.0 hálózatról van szó, akkor a 86.225.193.in-addr.arpa zónába bevezetünk 256 CNAME rekordot. Ezek jobb oldalán elvben tetszőleges domain név lehet, de szokás olyat megadni, ami az illető címtartomány kezdetét és nagyságát is jelzi, ilyenformán:

```
131.86.225.193.in-addr.arpa. CNAME 131.128/26.86.225.193.in-addr.arpa.
```

Az egyes kiosztott IP címtartomány darabokhoz megfelelően delegált in-addr.arpa-beli zónák tartoznak. Például a fenti esetben:

```
128/26.86.225.193.in-addr.arpa.     NS     ns.intezmeny.hu.
```

Ilyen módon a C osztályú címhez tartozó zónában a címek kiosztása után egyszer s mindenkorra rögzíteni lehet a bejegyzéseket, a kis címtartományt birtokló helyen pedig csak arra van szükség, hogy az inverz zóna neve c1.c2.c3.in-addr.arpa alak helyett cim/maszk.c1.c2.c3.in-addr.arpa alakú legyen. Ebbe a zónába aztán éppen úgy kell PTR rekordokat felvenni mintha teljes C osztályú címhez tartozna a zóna. Például:

```
131    PTR    bagoly.intezmeny.hu.
```

Ha ez a rekord a 128/26.86.225.193.in-addr.arpa. zónában van, akkor a fenti CNAME rekorddal együtt két lépcsőben feloldást ad a 131.86.225.193.in-addr.arpa domain névre, melynek eredménye bagoly.inetzmeny.hu.

BIND – Berkeley Internet Name Domain

BIND a neve az Interneten **leggyakrabban használt DNS implementációnak**. A program elsősorban unix típusú gépeken fut, de van pl. NT-s változata is. Fejlesztését az Internet Software Consortium támogatja, és a BIND „apukája”, Paul Vixie koordinálja. A BIND program forráskódban is szabadon letölthető az ftp.isc.org szerverről.

BIND változatok

E sorok írásakor a BIND kurrens változata 8.2.2. Használatosak azonban ennél sokkal régebbi változatok is. Jelentős ugrást jelentett 1998-ban a 4.9.x változatok után a 8.x változatok megjelenése. Ekkor a konfigurációs fájl szintaxisa is, a fő konfigurációs fájl neve is megváltozott.

4.9.x konfiguráció

Ezen változatok konfigurálásának archimédeszi pontja a `/etc/named.boot` fájl. Ebben kell leírni azt, hogy milyen zónákra elsődleges vagy másodlagos a szerver, és hogy a zónák milyen fájlokban tárolódjanak. Például:

```
;
tipus           domain                  source
;
primary         valahol.hu              valahol.zona
primary         1.2.199.in-addr.arpa    inverz.zona
secondary       amicus.hu                192.84.3.4      amicus.zona
```

Láthatjuk, hogy a primary kulcsszó után két paramétert kell megadnunk: a zóna nevét, és a fájlt, ami a zóna adatait tartalmazza. A secondary kulcsszóhoz három paraméter tartozik: a zóna neve, a név szerver(ek) ami(k)ről a zónát tükrözni kell, és a fájlnév, ahova a tükrözött adatokat mentjük.

A 4.9.x konfigurációk fontos sorai még:

```
directory   /var/named
cache       .     root.cache
```

A **directory** kulcsszó után megadott könyvtárhoz relatívan helyezkednek el a konfigurációban megadott fájlok.

A **cache** direktíva arra szolgál, hogy a gyökér névszerverek neveit és címeit tartalmazó állományt megadjuk. A névfeloldás – a szerver látó funkciója –, úgy fog zajlani, hogy ebből az állományból veszi a szerver a root szerverek adatait. A root szerverek száma egyre bővül. E sorok írásakor 13 névszerver autoritás az Interneten a root zónára. Fontos, hogy a root név szerverek listáját naprakészen tartsuk. A mindenkori lista megszerezhető a DNS segítségével. Ha például a host parancsot használjuk, akkor a következő parancs a friss.cache fájlba teszi az aktuális listát:

```
host -v -t ns -l . >friss.cache
```

A DNS gyökeréért felelős névszerverek listája letölthető ftp-vel is például az ftp.internic.net szerverről: ftp://ftp.internic.net/domain/named.root

8.x konfiguráció

A nyolcas változatú BIND-ok konfigurálásának archimédeszi pontja a `/etc/named.conf` fájl. Ennek sokkal bonyolultabb lehet a szerkezete, **árnyaltabb** feltételek megfogalmazására ad módot, mint egy 4.x konfiguráció. Ha valaki 4.x változatról 8.x-re akar áttérni, a `named.boot` fájl helyett az új szintaxisú `named.conf`-ra van szüksége. A 8-as disztribúció tartalmaz egy perl scriptet, ami **automatikusan konvertál** egy `named.boot`-ot `named.conf`-ra. Persze egy ilyen konfiguráció messze nem meríti ki az összes lehetőséget a lehetséges, és hasznos finomságokat illetően. Amint láttuk, a `named.boot/named.conf` fájl csak a kiindulási pontja a konfigurációnak: a tényleges DNS rekordok ettől különböző zóna fájlokban vannak. Ezek formátuma nem változott, nem is nagyon változhat: ami abban van, azt RFC írja le, nem implementációs kérdés.

A következőkben ismertetjük a 8.x konfiguráció néhány elemét.

Options utasítás

Ezzel az utasítással a konfiguráció egészére állíthatunk be tulajdonságokat, alapértelmezéseket. Példa:

```
Options
{
directory         "/usr/local/named";
named-xfer        "/usr/local/sbin/named-xfer";
notify            yes;
check-names       master fail;
check-names       slave warn;
```

```
check-names      response ignore;
recursion        yes;
transfers-in     20;
allow-transfer   {"amici";};
allow-query      {"anybody"};
listen-on        192.84.225.2;
}
```

A **directory** opció egy könyvtárra mutathat. A konfigurációban szereplő fájlnevek ehhez a könyvtárhoz relatívan értendők.

A **named-xfer** opciónak akkor van jelentősége, ha szerverünk slave (másodlagos) bizonyos zónákra. Ilyenkor a named-xfer opcióval annak programnak a nevét adhatjuk meg, amelyik a zónatranszfert végzi.

A **notify** egy új, és igen hasznos elem a 8. BIND-okban. Mód van arra, hogy a slave szerverek azonnal értesítést kapjanak, ha a zóna változott. Így nem kell kivárni amíg a SOA rekordban meghatározott „refresh" értéknek megfelelő időzítés lejár, a slave szerverek azonnal kezdeményezhetik a frissítést. Persze ez feltételezi, hogy a slave olyan BIND-ot futtat, ami ezt támogatja. Önmagában ez a tulajdonság is indokolja, hogy ne futtassunk 4.x változatú BIND-ot, hanem térjünk át frissebb változatra.

A **check-names** opció azt szabályozza, hogy ha a konfigurációs hibával szembesül a program, hogyan viselkedjen. Külön lehet szabályozni, mit tegyen, ha a saját master (elsődleges) zónáiban, a slave (másodlagos) zónákban, vagy egy kérdésre adott válaszban talál hibát. Mindegyik esetben három értéket lehet megadni:

- ignore Ügyet sem vet a hibára
- warn A logfájlba hibaüzenetet ír
- fail A logfájlba hibaüzenetet ír, és az adatot nem veszi figyelembe.

Ajánlatos legalább „master" zónákra beállítani a „fail" értéket. Így ha elrontjuk a zónafájlt, akkor szerverünk egyáltalán nem szolgáltatja, ezáltal hamar felfedezzük a hibát, és módunk lesz kijavítani.

A **recursion** opció azt határozza meg, hogy szerverünk csak láttat (az autoritatív zónákról nyújt információt), vagy hajlandó klienseknek kérdéseket megválaszolni a többi DNS szerverrel való kommunikáció révén. A TLD-ket szolgáltató szerverek általában nem rekurzívak.

A **transfers-in, transfers-out** opciókkal azt szabályozhatjuk, hogy egy időben hány zónatranszfer működhessen be, illetve kifele.

Az **allow-transfer** opcióval meghatározhatjuk, hogy kinek engedjük meg egész zónák elvitelét. Azoknál a zónáknál van ennek jelentősége, amelyekre autoritatívak vagyunk. Természetesen meg kell engedni a slave zónáknak, hogy a master-től elvigyék a zónát. Ezen kívül azonban megtilthatunk minden zónatranszfert. Régen, amikor nem volt olyan sok a visszaélés az Interneten, minden name szerverről le lehetett tölteni bárhova a zónákat. Ezen alapulnak például az internet host count-ok és statisztikák: az Internet bármely pontjáról meg lehet(ett) csinálni, hogy a root szerverektől kezdve bejárja egy program az egész név-fát, minden zónáról zónatranszfert kér, és például összeszámlálja az A rekordokat (ez a host count statisztika). Amikor ezek a sorok íródnak nagyon sok name szerver adminisztrátor korlátozza a zónatranszfert, és így kétes eredménnyel járna, akár csak a .hu alatti A rekordok összeszámlálása is. Az allow-transfer opció paramétere egy acl (access control list), egy címlista. A címlistában hálózatokat, IP címeket sorolhatunk fel ilyenformán:

```
Acl
amici {
  199.2.2.4;
```

```
    192.3.4.5;
    195.2.3/24;
};
```

Az első két sor két IP címről – a slave-ekről –, engedi meg a zónatranszfert. A harmadik sorban a /24 egy bitmaszkot jelent: a 193.2.3. hálózat bármely gépének megengedjük a zónatranszfert. Az így definiált lista nevére aztán hivatkozhatunk más utasításokban (pl. options/allow-transfer, zone). Az acl utasításnak azonban meg kell előznie a névre való hivatkozást.

Az **allow-query** opcióval korlátozhatjuk, hogy honnan jövő kéréseket válaszoljon meg szerverünk. Paramétere ennek is egy címlista.

Az options utasításban megadott „allow-transfer” és „allow-query” opciót az egyes zónák definiálásánál hasonló szintaxisú utasítással felülbírálhatjuk.

A **listen-on** opciónak akkor van jelentősége, ha több IP címe van szerverünknek: meghatározhatjuk, hogy melyik IP címekre érkező kérdésekre válaszoljon. Alapértelmezésben mindegyiken elérhető a DNS szerver.

A 4.x konfigurációnál használt „primary” és „secondary” és „cache” utasításokat a **zone** utasítás váltotta fel. Példák:

```
zone valami.hu  {
type  master;
file  "valami.hu";
allow-transfer  {"amici";};
};

zone mas.hu {
type slave;
masters { 192.2.3.4;
        193.4.3.2;
        }
file "sec/mas.hu";
}

zone "." {
type hint;
file "root.hints";
};
```

A valami.hu-ra elsődleges (master) a mas.hu-ra másodlagos (slave) szerverek vagyunk. A valami.hu zónát a valami.hu fájlban kell prezentálnunk a szervernek. A mas.hu zónát a szerver a sec alkönyvtár mas.hu nevű fájljába kérjük. A valami.hu zónánál felülbíráltuk az „options”-ban megadott alapértelmezését a zónatranszfer engedélyezésnek: itt csak az „amici” nevű címlista tagjainak engedjük meg az átvitelt. A mas.hu zónánál két „master” szervert is felsoroltunk. Ezek közül értelemszerűen (legalább) az egyik szintén „slave”, vagy ugyanannak a „master” szervernek két különböző IP címéről van szó. A „masters” utasításban megadott IP címek sorrendje prioritást jelent: elsősorban az elsőről szinkronizálja a szerver a zónát, ha az nem sikerül, akkor megy tovább a listán. Általában nem érdemes több IP címet megadni, de hasznos lehet, ha komoly esély van arra, hogy az elsőről nem sikerül szinkronizálni. A gyökér szerverekre vonatkozik a „hint” típusú zóna. Az itt megadott fájl-ba (root.hints) tegyük a root name szerverekre vonatkozó NS és A rekordokat.

Hasznos segédprogramok

A DNS adatok lekérdezésének klasszikus eszköze az nslookup. Ez a program szinte minden operációs rendszernek szabványos része. Vannak azonban szabadon terjeszthető alternatívái, amiknek talán kényelmesebb a felhasználói felülete, és a DNS kérdezésen kívül sok mást – például hibafelderítést – is közvetlenül támogatnak.

Host

Ennek a programnak több változata is elterjedt. Igen jó, sokat tudó és folytonosan fejlődő az Erik Wassenar féle. Kurrens változata letölthető az ftp.nikhef.nl-ről. Érdemes letölteni és installálni az friss változatot akkor is, ha operációs rendszerünk már tartalmazná a program valamely változatát. A BIND disztribúcióval is jön egy változat, de általában az sem elég friss.

Dig

A Dig hasonló célokat szolgál, mint a host. Ízlés és szokás kérdése, hogy ki melyiket használja.

Grafikus DNS adminisztrációs segédletek

Felmerül az igény, hogy ne kézzel editáljuk a DNS konfigurációs fájlokat, hanem pl. egy grafikus felületen át. Erre több megoldás is készült. Egyik ezek közül a Webmin. Ez több rendszeradminisztrációs feladatra – többek közt DNS adminisztrációra – alkalmas web-es felületet nyújtó eszköz. Otthona: http://www.webmin.com/webmin/

Megjegyzendő, hogy az ilyen eszköz nem pótolja a hozzáértést.

Regisztrálás a .hu TLD és a .hu alatti közcélú SLD-k alatt

A .hu alatti regisztrálás feltételeit az Internet Szolgáltatók Tanácsa szabályozza. Létrejött néhány közcélú második szintű (SLD) domain: org.hu, co.hu, info.hu stb. Az ezek alatti és a .hu alatti domain név igénylésről részletes tájékoztató olvasható a www.nic.hu web lapokon. A lényeg az, hogy van egy sor szolgáltató, akiknél egyformán be kell tartani néhány formai és technikai szabályt, de a szolgáltatás díját a szolgáltatók önállóan, egymással versenyben állapítják meg.

Az egyik formai szabály, hogy .hu alatt csak olyan nevet lehet regisztrálni, ami az igénylő neve, nevének rövidítése vagy valamilyen védjegye.

Két fő technikai szabály:

- A bejegyzendő domaint legalább két, független hálózati elérésű domain név szerverrel kell szolgáltatni.
- A postmaster@domain.hu levelezési címnek – ahogy azt az RFC822 is előírja – működnie kell.

Példák

Néhány példa konfigurációs fájl következik, a sorok között magyarázatokkal. Named.conf fájl csak látó, nem láttató (caching-only) szerver számára:

```
options {
          directory "/usr/local/named";
          recursion yes;
};

zone "." {
          type hint;
          file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
          type master;
          file "127.0.0";
};
```

Named.conf láttató szerver számára:

```
/*
   Elsődleges a valahol.hu és a 193.111.222
   C osztályú címtartomány inverze számára, másodlagos
```

```
   a mas.hu zóna számára. Szükséges fájlok még:
   root.hints  -  a root szerverek adataival
   valahol.hu  -  a szükséges NS, A, stn. Rekordokkal
   193.111.222 -  a szükséges PTR rekordokkal
   127.0.0     -  a loopback hálózat számára
*/

options {
          directory "/usr/local/named";
          recursion yes;
          notify yes;
          check-names slave warn;
          check-names master fail;
          allow-transfer {"slaves";};
};

/* csak két címről engedünk meg zónatranszfert: */

acl "slaves" {
  193.222.111.1;
  193.111.222.5;
};

zone "." {
          type hint;
          file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
          type master;
          file "127.0.0";
};

zone "valahol.hu" {
          type master;
          file "valahol.hu";
};

zone "222.111.193.in-addr.arpa" {
          type master;
          file "193.111.222";
};

zome "mas.hu" {
          type slave;
          masters { 193.111.222.5;}
          file "sec/mas.hu";
};
```

A 0.0.127.in-addr.arpa zónára a következő fájl elegendő:

```
@            SOA     ns.valahol.hu. hostmaster.valahol.hu. (
                     1999091001       ; Serial
                     86400    ; Refresh
                     7200     ; Retry
                     604800   ; Expire
                     86400)   ; Minimum TTL
             NS      ns.valahol.hu.

1            PTR     localhost.
```

A valami.hu fájl lehet ilyen:

```
@           SOA     ns.valahol.hu. hostmaster.valahol.hu. (
                    1999091001       ; Serial
                    86400   ; Refresh
                    7200    ; Retry
                    604800  ; Expire
                    86400)  ; Minimum TTL
            NS      ns.valahol.hu.
            NS      ns.mas.hu.
ns          MX      10 masina.valahol.hu.
masina      A       193.111.222.1
ns          A       193.111.222.3
www         CNAME   masina
```

A 193.111.222 fájl lehet ilyen:

```
@           SOA     ns.valahol.hu. hostmaster.valahol.hu. (
                    1999091001       ; Serial
                    86400   ; Refresh
                    7200    ; Retry
                    604800  ; Expire
                    86400)  ; Minimum TTL
            NS      ns.valahol.hu.
            NS      ns.mas.hu.
1           PTR     ns.valahol.hu.
3           PTR     masina.valahol.hu.
```

Források az Interneten

DNS-sel kapcsolatos RFC-k. Elérhetők például itt: ftp://ftp.sztaki.hu/pub/nic/rfc

- rfc974, Mail routing and the domain system
- rfc1032, Domain administrators guide
- rfc1033, Domain administrators operations guide
- rfc1034, Domain names – concepts and facilities
- rfc1035, Domain names – implementation and specification
- rfc1101, DNS encoding of network names and other types
- rfc1122, Requirements for Internet hosts – comm. layers
- rfc1123, Requirements for Internet hosts – application
- rfc1536, Common DNS implementation errors
- rfc1537, Common DNS data file configuration errors
- rfc1591, Domain Name System structure and delegation
- rfc1597, Address allocation for private internets
- rfc1627, Network 10 considered harmful
- rfc1700, Assigned numbers
- rfc1706, DNS NSAP resource records
- rfc1712, DNS encoding of geographical location (GPOS)
- rfc1713, Tools for DNS debugging
- rfc1794, DNS support for load balancing

- rfc1876, Expressing location information in the DNS (LOC)
- rfc1884, IP v6 addressing architecture
- rfc1886, DNS extensions to support IP v6 (AAAA)
- rfc1912, Common DNS operational and configuration errors
- rfc1982, Serial number arithmetic
- rfc1995, Incremental zone transfer in DNS (IXFR)
- rfc1996, Prompt notification of zone changes
- rfc2010, Operational criteria for root nameservers
- rfc2052, Specification of location of services (SRV)
- rfc2065, DNS security extensions (KEY/SIG/NXT)
- rfc2317, Classless IN-ADDR.ARPA delegation

A bind lapjai: http://www.isc.org

A Bind Operations Guide, a BOG. Része a Bind disztribúciónak, de letölthető külön is: http://www.dns.net/dnsrd/docs/bog

DNS-sel kapcsolatos információk: http://www.dns.net

Kitűnő és olvasmányos könyv a DNS-ről: Cricket Liu & Paul Albitz: DNS and BIND O'Reilly kiadó

A .hu alatti regisztrálásról: http://www.nic.hu

A linux disztribúciók DNS HOWTO-ja minden disztríbúciónak része. Letölthető pl. innen: ftp://ftp.kfki.hu/pub/linux/sunsite.unc.edu/docs/HOWTO/DNS-HOWTO